Pronunciation and Accent Corrector

A **Project Report** submitted in partial fulfillment of
the requirements for the degree of

Bachelor of Engineering

by

Saurabh Deshmukh	**Roll No.-12**
Akash Doifode	**Roll No.-16**
Rajat Hudge	**Roll No.-18**
Vyomkesh Shete	**Roll No.-63**

Under the guidance of
Prof. M. V. Kamble

DEPARTMENT OF ELECTRONICS ENGINEERING
VISHWAKARMA INSTITUTE OF TECHNOLOGY PUNE
2013 - 2014

Chapter 1

Introduction

Pronunciation or Accent plays important role in identifying the individuals country, religion, region, cast, community etc. There are various aspects of pronunciation out of which one is about people having stammering problem. People with this problem cant produce word or sentence with correct pronunciation. To counter this problem we are designing a system named Pronunciation and Accent Corrector. There are also issues with employees which are working in the BPO, calling centre, customer care about their pronunciation. System is mainly divided into three parts; speech to text conversion and text to speech conversion, Auto-suggestion Graphical Interface.

In speech to text conversion, system starts with pre-processing stage, which takes speech waveform from stammering people as its input and extracts feature vectors from it or observations which represent the information required to perform recognition. The MFCC bank for speech recognition is already stored In database.AS stammering people cant pronounce the word know which word they want to produce, so from part of word they pronounce auto suggested word will be displayed on screen. From that word user can select required word and word will pass In his voice using text to speech system.

In Text to speech system, users voice samples of few words are taken in learning process. And from that word we extract group of phonemes and using phonetic concatenation (concatenation of group of phonemes) and speech synthesis is done. By this way, stammering people can produce a word or sentence with correct pronunciation.

Chapter 2

Literature Survey

2.1 Methods of Speech Synthesis

2.1.1 Articulatory Synthesis

Articulatory Synthesis tries to model the human speech production system directly by modelling the motion of various articulators involved in the speech production. It is based on the articulatory theory of speech production. Various physical articulators like tongue, lips, velum, nose, pharynx, vocal cord etc. are modelled directly in this method. In an articulatory model the tube corresponding to the vocal tract is usually divided into many small sections, and each section is approximated by an electrical transmission line. Typical Articulatory Synthesis system uses 7-11 parameters to adequately describe motion of various articulators.

The synthesizer requires one parameter for controlling velum opening, one for lip rounding, one for lip closure, two each for the tongue body and tongue tip as they have both vertical and horizontal degrees of freedom, one each for jaw height, pharynx width, and larynx height. Coarticulation constraints and motion constrains of articulator are considered while calculating the values of these parameters. These parameters values are stored for each phoneme along with their time sequences to accurately synthesize each phoneme.

Each phoneme is produced by motion of some crucial articulators modelled by important parameters and some unimportant articulators modelled by unimportant parameters. e.g. in production of /m/ lip and velum closure are important parameters, while tongue configuration has only little importance. Hence it becomes very important to model and control the crucial articulator parameters.

Another approach to model vocal tract models the area function rather than moving of actual articulators. The approach models around 11 cross-sectional areas like glottal opening area, supraglottal constriction area etc. The vocal tract area models allow accurate modelling of transients due to abrupt area changes, as well as automatic generation of turbulence excitation for narrow constrictions. But then the model loses the easy coarticulation description of the previous model.

Experimental data describing the motion of articulators is derived from the motion of articulators is derived from X-rays or MRI (magnetic resonance imaging) of vocal tract during simple utterances. Earlier in 1960s, only X-ray technique was used to capture the motion of articulator in vocal tract as MRI technique was in its infancy. X-ray images showed only two dimensional model of the three dimensional vocal tract, also little data was available due to danger of X-ray exposure, hence the modelling of Articulatory Synthesis was limited by availability of data. Nowadays, with advancement in MRI technique, detailed information is available to model the motion of articulators in vocal tract. With this data availability and advent of computers, there have been significant advances in modelling of articulators and the acoustic processes in the vocal tract. But, the considerably greater computational cost of Articulatory Synthesis makes the use of these models difficult in practical systems.

2.1.2 Concatenative Synthesis

The simplest method to synthesize speech would be to store different words generally spoken beforehand in machines memory and then retrieve the required words from the memory and simply concatenate them to produce the sentence to be spoken. But as the number of words that are used in different literatures is extremely large, storing all of those will be quite impossible. Moreover new words are coined every day. Also such an approach will fail as a spoken sentence is very different from a sequence of words uttered in isolation.

Thus for avoiding the necessity of storing large number of words for synthesis of unrestricted text, basis sounds (phonemes) can be used, as basis sounds can be combined together to from words and hence sentences. Most of the languages have only 30-40 phonemes, so storing the phonemes can greatly reduce memory requirements. But the sound synthesized using this approach is much slower compared to natural speech and also has discontinuities at the point of concatenation of phonemes i.e. phoneme boundaries. The pronunciation of a phoneme in a phrase is also heavily affected by the neighbouring phoneme, intonation (variation of pitch while speaking) and speaking rate. Hence phoneme concatenation will not produce good quality natural sound.

Thus from the above two examples of word concatenation and phoneme concatenation it is clear that one of the most important aspects in concatenative synthesis is to find correct speech unit length which can be used for concatenation. In present systems unit used usually words, syllables, demisyllables, phonemes, diphones, and sometimes even triphones.

Syllables are considered as the phonological building blocks of words. For example, the word water is composed of two syllables: wa and ter. A syllable is typically made up of a syllable nucleus (most often a vowel) with optional initial and final margins (typically, consonants). The syllable can be a promising unit for concatenation, but there are over 10000 different syllables in English.

Demisyllables are speech units obtained by cutting syllables into half, with the cut in the middle of the vowel. Cutting the syllable at the middle of vowel is preferred to rapid transitions at phone boundaries as the effects of coarticulation are minimal at the middle of vowel. As we are using bigger concatenation unit, speech synthesized using Demisyllables concatenation requires considerably less concatenation points compared to phonemes and diphones. With purely demisyllables system, some proper names cant be synthesized properly. As an example of word formation using concatenation of demisyllables consider the word straight. To synthesize the word straight two demisyllable sequence $/stre - et/$ would be used $(denotessilence)$.

Diphones are obtained by cutting a speech waveform into phone-sized units, with a cut in the middle of each phone so as to preserve the transition between adjacent phones in each diphone. Coarticulatory influences are observed to be minimal at the middle of a phoneme. Speech synthesized with diphone concatenation has less distortion at concatenation points and has good coarticulation effect. There are thus about 40 times 40, or 1600, different diphone possibilities. It will be also necessary to include several different versions of each diphone to distinguish between stressed and unstressed syllables and to include allophones. Even if we store all these diphones, it has been observed that it would be difficult to change the duration and fundamental frequency contour on a diphone to vary the pitch of utterance. Also a disadvantage of the diphone approach is that discontinuities may appear right in the middle of vowels if the two adjoining diphones do not reach the same vowel target as in bill and wet. As an example of word formation using concatenation of diphones again consider the word straight. To synthesize the word straight, the six-diphone sequence $/s - st - tr - re - et - t/$ would be used $(denotes silence)$.

Problems in Concatenative Synthesis

1) Un-natural synthetic speech due to discontinuities at concatenation points.

2) Memory requirements are usually very high when long concatenation units are used.

3) Data collecting and labelling of speech samples is usually time-consuming.

4) Modelling the synthetic speech matching different individuals traits is quite difficult.

2.1.3 Formant Synthesis

It is based on the acoustic theory of speech production. It is a source-filter model of speech production. Excitation produced by the source passes through the filter (modelling the vocal tract), is modified by the resonance characteristics of the filter to produce speech. The source is either modelled by an impulse train for voiced speech or a pseudo-random noise for unvoiced speech. The vocal tract is usually modelled by a cascade of resonators. In a simple model, transfer function of the linear filter modelling the vocal tract has only poles.

The primary sources of sound are voicing, produced by the vibration of the vocal cord, and turbulence noise produced by pressure variation across the constriction formed in the vocal tract. The acoustic tube (vocal tract) modelled as linear filter, is formed by pharynx, oral cavity and lips. Each complex conjugate pair of pole in the transfer function of filter produces a local peak in the spectrum, known as formant. So to model first four formants and 2 poles of source, 10 poles are required. For nasal, fricative and nasalized vowel sounds, some zeros should be added to the transfer function to model the sound absorbing properties of the side-branch tube in complex articulations.

From above three methods, Concatenative Synthesis is simple and easier to implement as compared to other two. So, we have used Concatenative Synthesis method for speech synthesis in our project.

Speech Synthesis Using Concatenative Synthesis

A text to speech (TTS) synthesizer is a computer based system that can read text aloud automatically. A speech synthesizer can be implemented by both hardware and software. It has been made a very fast improvement in this field over the couple of decades and lot of high quality TTS systems are now available for commercial use. Speech is often based on concatenation of natural speech i.e units that are taken from natural speech put together to form a word or sentence. Concatenative speech synthesis has become very popular in recent years due to its improved sensitivity to unit context over simpler predecessors. Rhythm is an important factor that makes the synthesized speech of a TTS system more natural and understandable; the prosodic structure provides important information for the prosody generation model to produce effects in synthesized speech. Many TTS systems are developed based on the principle, corpus-based speech synthesis. It is very popular for its high quality and natural speech output.

Architecture of Text to Speech

The TTS system comprises of these 5 fundamental components:

A. Text Analysis and Detection

B. Text Normalization and Linearization

C. Phonetic Analysis

D. Prosodic Modelling and Intonation

E. Acoustic Processing

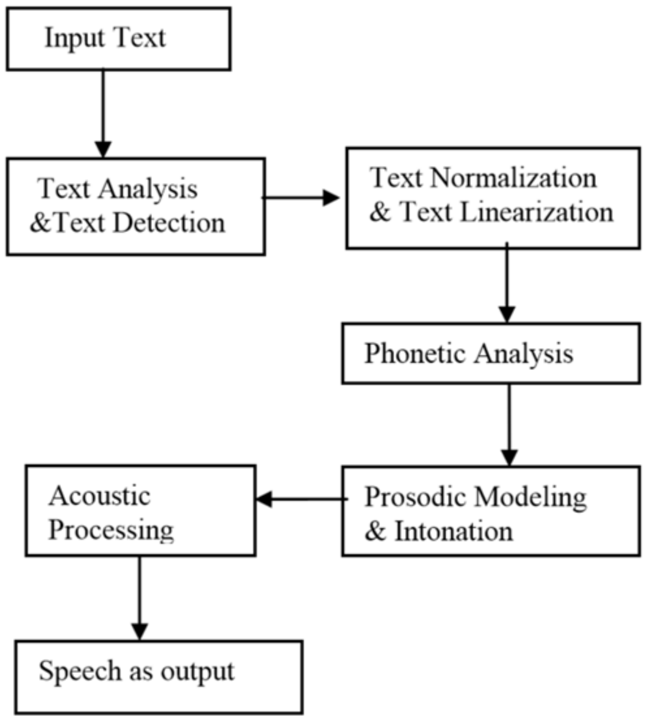
Input Text
Text Analysis &Text Detection
Text Normalization & Text Linearization
Phonetic Analysis
Prosodic Modeling & Intonation
Acoustic Processing
Speech as output

Figure 2.1:

The input text is passed through these phases to obtain the speech.

A. Text Analysis and Detection

The Text Analysis part is pre-processing part which analyse the input text and organize into manageable list of words. It consists of numbers, abbreviations, acronyms and idiomatic and transforms them into full text when needed. An important problem is encountered as soon as the character level: that of punctuation ambiguity (sentence end detection). It can be solved, to some extent, with elementary regular grammars.

Text detection is localizing the text areas from any kind of printed documents. Most of the previous researches were concentrated on extracting text from video. We aim at developing a technique that work for all kind of documents like newspapers, books etc.

B. Text Normalization and Linearization

Text Normalization is the transformation of text to pronounceable form. Text normalization is often performed before text is processed in some way, such as generating synthesized speech or automated language translation. The main objective of this process is to identify punctuation marks and pauses between words. Usually the text normalization process is done for converting all letters of lowercase or upper case, to remove punctuations, accent marks , stopwords or too common words and other diacritics from letters .

Text normalization is useful for example for comparing two sequences of characters which represented differently but mean the same. $ontvsonot, Imvs.Iam, Cantvs.cannot$ are some of the examples.

C. Phonetic Analysis

Phonetic Analysis converts the orthographical symbols into phonological ones using a phonetic alphabet. Basically known as $grapheme - to - phoneme$ conversion.

Phone is a sound that has definite shape as a sound wave. Phone is the smallest sound unit. A collection of phones that constitute minimal distinctive phonetic units are called Phoneme. Number of phonemes is relatively smaller than the graphemes, only 44. Phoneme Set (English),

Vowels (19) : $/a/, /ae/, /air/, /ar/, /e/, /ee/, /i/, /ie/, /o/, /oe/, /oi/, /oo/, /ow/, /or/, /u/, /ur$

Consonants (25) : /b/, /ks/gz/, /c/k/, /ch/, /d/, /f/, /g/, /h/, /j/, /l/, /m/, /n/, /ng/, /p/, /kw/, /r/, /s/, /sh/, /t/, /th/, /th/, /v/, /y/, /z/, /zh/.

Examples:

o /air/ : square, bear.

o /ow/ : down, house.

o /ks/gz/ : box, exist

Pronunciation of word based on its spelling has two approaches to do speech synthesis namely,

(a) Dictionary based approach:

There are 44 phonemes in English language were taken into account. The selection of

44 phonemes were based on the list from the source namely Orchestrating Success in Reading by Dawn Reithaug (2002) under National Right to Read Foundation. The task was to extract out the sounds pertaining to these 44 phonemes which forms as a base for creating any English word in a standard lexicon. The samples of sounds imported into the Matlab workspace are stored in a column vectors. All samples of the imported sound contribute to a particular sound intensity depending on its amplitude, hence, power. The combination of all these components yields the total sound (of the phoneme). So, a few consecutive samples could be selected and then stored in another vector and played using the $sound$ command. Therefore, by hit and trial method various desired sounds (phonemes) could be extracted from the imported $.wav$ files.

Sample Code:

$[y, fs1, ns1] = wavread('pain.wav');$

yAIN=y(12000:1:16000);

(b) Rule based approach.

A dictionary is kept were It stores all kinds of words with their correct pronunciation; its a matter of looking in to dictionary for each word for spelling out with correct pronunciation. This approach is very quick and accurate and the pronunciation quality will be better but the major drawback is that it needs a large database to store all words and the system will stop if a word is not found in the dictionary.

The letter sounds for a word are blended together to form a pronunciation based on some rule. Here main advantage is that it requires no database and it works on any type of input. Same way the complexity grows for irregular inputs

D. PROSODIC MODELLING AND INTONATION

The concept of prosody is the combination of stress pattern, rhythm and intonation in a speech. The prosodic modelling describes the speakers emotion. Recent investigations suggest the identification of the vocal features which signal emotional content may help to create a very natural [9] synthesized speech.

Intonation is simply a variation of speech while speaking. All languages use pitch, as intonation to convey an instance, to express happiness, to raise a question etc. Modelling of an intonation is an important task that affects intelligibility and naturalness of the speech. To receive high quality text to speech conversion, good model of intonation is needed.

Generally intonations are distinguished as (I) Rising Intonation (when the pitch of the voice increases) (ii) Falling Intonation (when pitch of the voice decreases) (iii) Dipping Intonation (when the pitch of the voice falls and then rises) (IV) Peaking Intonation (when the pitch of the voice raises and then falls)

E. Acoustic Processing

The speech will be spoken according to the voice characteristics of a person, there are three type of Acoustic sintering available,

(I).Concatenative Synthesis (ii).Formant Synthesis (iii).Articulator Synthesis

The concatenation of pre-recorded human voice is called Concatenative synthesis, in this process a database is needed having all the pre-recorded words. The natural sounding speech is the main advantage and the main drawback is the using and developing of large database.

Formant-synthesized speech can be constantly intelligible. It does not have any database of speech samples. So the speech is artificial and robotic.

Speech organs are called Articulators. In this articulator synthesis techniques for synthesizing speech based on models of the human vocal tract are to be developed. It produces a complete synthetic output, typically based on mathematical models.

Now, the phonemes can be concatenated to give various words. Since, all these sounds (phonemes) are just column vectors, their constituent elements could be placed one after another and stored in another variable (vector). This is concatenation.

This way all the words could be played by merely selecting the phonemes and placing the phoneme vectors one after another.

Sample Code:

$[y, fs1, ns1] = wavread('pain.wav');$

yAIN=y(12000:1:16000);

[z,fs1,ns1]=wavread('gain.wav');

zG=z(150:1:950);

new=[zG;yAIN];

In this code, the sound $AINfrompain$ is extracted and stored in yAIN and the sound $Gfromgain$ is extracted and stored in zG. Therefore, to obtain the sound of $gain$ using these two variables we perform concatenation as shown above.

Thus, after this stage a dictionary of words (with $.wav$ extension) has been created and stored for future reference. The phonemic representation of all words is also stored in this dictionary. For example, phonemic representation of $girlisg/ur/l$. It helps in separating the phonemes of the whole word. The dictionary gets stored in the matlab directory as a text file.

The dictionary can be accessed for desired comparison with the user input by using the following code

Sample Code:

$fid = fopen('trail.txt')$

trail.txt$is the dictionary for the program. The user interested to hear the speech version (sound) of an$

The input text file (dlm1.txt in the above program) can be imported using the Matlab code as shown below:

Sample Code:

$x = textread(dlm1.txt, This code imports the text file into the Matlab workspace$

2.2 Methods of Speech Recognition

2.2.1 Feature Extraction

Cepstral Analysis

This analysis technique is very useful as it provides methodology for separating the excitation from the vocal tract shape. In the linear acoustic model of speech production, the composite speech spectrum, consist of excitation signal filtered by a time-varying linear filter representing the vocal tract shape.

Mel-Cepstrum Analysis

This analysis technique uses cepstrum with a nonlinear frequency axis following mel scale. For obtaining mel cepstrum the speech waveform s(n) is first windowed with analysis window w(n) and then its DFT S(k) is computed. The magnitude of S(k) is then weighted by a series of mel filter frequency responses whose center frequencies and bandwidth roughly match those of auditory critical band filters.

The next step in determining the mel cepstrum is to compute the energy in this weighted sequence. If $V_1(k)$ is the frequency response of l^{th} mel scale filter. The resulting energies are given for each speech frame at a time n and for the l^{th} mel scale filter are,

$E_{mel}(n,l) = (\frac{1}{A_l}) \sum k = L_l{}^{U_l} |V_l(k)S(k)|^2$

Where $U_l and L_l$ are upper and lower frequency indices over which each filter is nonzero and A_l is the energy of filter which normalizes the filter according to their varying bandwidths so as to give equal energy for flat spectrum.

The real cepstrum associated with $E_{mel}(n,l)$ is referred as the mel-cepstrum and is computed for the speech frame at time n as

$C_{mel}(n,m) = (\frac{1}{N})\sum l = 0^{N-1} logE_{mel}(n,l)cos[\frac{2\Pi(l+\frac{1}{2}}{)}N]$

Such mel cepstral coefficients C_{mel} provide alternative representation for speech spectra which exploits auditory principles as well as decorrelating property of cepstrum.

Linear Prediction

The main idea behind prediction is to extract the vocal tract parameters. Given a speech sample at time n,s(n) can be modelled as a linear combination of the past p speech samples,

$s(n) = b_0u(n) + a_1s(n-1) + a_2s(n-2) + ... + a_ps(n-p)$

$s(n) = b_0u(n) + \sum i = 1^p a_is(n-i)$

Where u(n) is a normalized excitation signal, b_0 the gain of the excitation signal and the coefficients $a_1, a_2..a_p$ are the weights for previous sound samples. All these coefficients are assumed constant over the speech analysis frame. An other way to look at this is to express s(n) in the z-domain:

$S(z) = b_0U(z) + \sum i = 1^p a_iS(z)z^{-i}$

And hereby the transfer function becomes:

$H(z) = \frac{S(z)}{U(z)} = \frac{b_0}{1-\sum i=1^p a_iz^{-i}}$

H(z) shows ,one can see that this problem is to create an all pole model of the vocal tract.The calculation of the coefficients are applied when the speech is assumed stationary, $x_2(k;m)$ are the frames of speech where the speech is assumed stationary.The calculation of these coefficients for each block in $x_2(k;m)$ can be done in different ways by using the auto-correlation method, the covariance method, the Levinson-Durbin recursion etc.

2.2.2 Training and Testing

Hidden Markov Model

As mentioned in the introduction part the technique used to implement the speech recognition system was the Hidden Markov Model, HMM. The technique is used to train a model which in our case should represent a utterance of a word. This model is used later on in the testing of a utterance and calculating the probability of that the model has created the sequence of vectors.

The difference between an Observable Markov Model and a Hidden Markov Model is that in the Observable the output state is completely determined at each time t. In the hidden

Markov Model the state at each time t must be inferred from observations. An observation is a probabilistic function of a state. For further information about the difference and information about the Observable Markov Model and Hidden Markov Model please refer to [MM].

The hidden Markov Model is represented by $\lambda = (\pi, A, B)$.

π= initial state distribution vector.

A= State transition probability matrix.

B = continuous observation probability density function matrix.

Training

Given a N number of observation sequences of a word $O_N = o_1 o_2 o_3 .... o_T$. How is the training of that model done to best represent the word. This is done by adjusting the parameters for the model $\lambda = (\pi, A, B)$. The adjustment is an estimation of the parameters for the model $\lambda = (\pi, A, B)$ that maximizes P(O—λ).

Testing

When comparing an observation sequence O = $o_1 o_2 o_3 .... o_T$ with a model $\lambda = (\pi, A, B)$ you need to find the solution to problem two . The solution is about finding the optimal sequence of states q = $q_1 q_2 q_3 .... q_T$ to a given observation sequence and model. There is different solutions depend on what is meant by optimal solution. In the case of most likely state sequence in its entirety, to maximize P (q—O, λ) the algorithm to be used is the Viterbi Algorithm, state transition probabilities has been taken into account in this algorithm, which is not done when you calculate the highest probability state path. Due to the problems with the Viterbi Algorithm, multiplication with probabilities, The Alternative Viterbi Algorithm is used. The testing is done in such matter that the utterance to be tested is compared with each model and after that a score is defined for each comparison.

Chapter 3

Work Done

3.1 Speech to Text

In speech to text conversion, input is given to pre-processing stage.And then given to frame blocking and windowing section. After doing windowing resultant speech is given to feature extraction, which mainly uses mfcc coefficients. And then finally given to the post processing section. In this way we can extract the features from speech signal.

3.1.1 Pre-Processing

This step is the first to create feature vector. The objective in the pre-processing is to modiey the speech signai, x(n), so that it will be more suitable for feature extraction analysis. The pre-processing operations such as noise cancelling, preemphasis and voice activation detection(VAD) are explained below.

i)Noise Cancellation

The first thing to consider is if the speech, x(n), is corrupted by some noise d(n), for example an additive disturbance x(n)=s(n)+d(n), where s(n) is a clean speech signal. Two commonly used noise reduction algorithms in field of speech recognition context is spectral substraction and adaptive noise cancellation. A low signal to noise ratio decrease the performance of the recognizer in a real environment.

ii)Preemphasis

The preemphasizer is used to spectrally flatten the speech signal.This is usually done by a highpass filter. The most commonly used filter for this step is FIR filter described below,

$H(Z) = 1 - 0.95z^{-1}$

The filter in the time domain will be, h(n)=1, -0.95 and filtering in the time domain will give the preemphasized signal s1(n):

$s_1(n) = \sum k = 0^{M-1} h(k)s(n-k)$

iii) Voice Activation Detection(VAD):

The problem of locating the end points of an utterance in aspeech signal is amajor problem for the speech recognizer. An inaccurate end point detection will decrease the performance of the speech recognizer. The problem of detecting end points seem to be relatively trivial, but it has been found very difficult in practice. Some commonly used measurements for finding speech are short term energy estimate Es1, or short term power estimate Ps1, nad short term zero crossing rate Zs1.

For each block of L samples these measures calculate some value. Note that the index for this function is m and not n, this because these measures do not have to be calculated for every sample(we have calculated measures in every 20ms). The short term zero crossing rate gives a measure of how many times the signal, s1(n), changes sign. This short term zero crossing rate tends to be large during unvoiced regions.

These measures will need some triggers for making decisions about where the utterances begin and end. This done by assuming that the first 10 blocks are backgrounding noise. With this assumption the mean and the variance for the measures wil be calculated. To make a comfortable approach the following function is used:

$W_{s1}(m) = P_{s1}(m).(1 - Z_{s1}(m)).S_c$

Using this function zero crossing rate will be taken into account. Here Sc=1000, which is a scale factor for avoiding small values. The trigger for this function can be described as,

$t_w = \mu_w + \alpha\delta_w$

The μW is the mean and W is variance for Ws1(m) calculated for the first 10 blocks. The α term is a constant that have to be fine tuned according to the charachteristics of the signal. After some testing the following approximation of α will give a pretty good voice activation detection in a various lavel of additive background noise.

$\alpha = 0.2.\delta_w^{-0.8}$

The VAD(m) function can now be found as,

$VAD(m) = 1, W_{s1}(m)t_w$

=0, W$_{s1}(m) < t_w$

VAD(m) is found of VAD(m) in the block of measure. For example if the measure is calculated every 320 sample(block length L=320), which correspond to 20ms if the sampling rate is 16kHz.

3.1.2 Windowing And Frame Blocking

The next thing to do with x1(n) is to divide it into speech frames and apply a window to each frame. Each frame is a k samlples long, with adjacent frames being separated by p samples. Typically values of k and p are 320 samples and 100 samples(62.5

$w(k) = 0.54 - 0.46cos\frac{2\Pi k}{K-1}$

3.1.3 Feature Extraction

In feature extraction there consist of two parts: the cepstrum calculation and a method called mel scaling. The parts will be described in the following two sections and finally mel-cepstrum will be summarized.

Cepstrum:-

The cepstrum method is a way of finding the vocal tract filter H(z) with $homomorphicprocessing$ .Homomorphic signal processing is generally concerned with the transformation to linear domain of signals obtained in a nonlinear way. In this case the two signals are not combined linearly (a convolution cant be described as an simple linear combination). The speech signal can be seen as the result of a convolution between u(n) and h(n):

$S(n) = b_0.u(n)(n)$

In frequency domain:

$S(z) = b_0.U(z)H(z)$

Since the excitation U(z), and the vocal tract, H(z) ,are combined multiplicative it is difficult to separate them. If the log3 operation is applied the task will become additive:

$logS(z) = log(b_0.U(z)H(z)) = log(b_0U(z)) + log(H(z))$

The additive property of log spectrum also applies when inverse transforming is applied to it. The Result of this operation is called a cepstrum. To avoid taking logarithm of complex numbers , an abs Operation is applied to S(z), this is the definition of a $realcepstrum.$

Note that the real cepstrum is an even sequence on the index n, since $C_s(z) = logS(z)$, is real and even. These properties gives that the inverse cosine transform can be applied to

$C_s(z)$ to get $c_s(n)$.The index n in $c_s(n)$ is now the so called quefrency high-quefrency equals high n and vice versa. Now that $c_s(n)$ has been calculated one will get hold of $c_h(n)$, which is the cepstrum of the vocal tract filter.the vocal tract filter has $slow$ spectral variations and the excitation signal has $fast$ variations. This propert corresponds to low-quefrency for the voacal tract filter and high quefrency domain operations.Note that the P in the cepstrum for the excitation is the corresponding pitch.

To extract the vocal tract cepstrum one can apply a low-pass $lifter$ to $c_s(n)$ is to be multiplied with:

$l_1(n) = 1, n = 0, 1, ..., L - 1$

$= 0, else$

where the length ,L,is the chosen to extract $c_h(n)$. A good choice of L would be 75. An other lifter which has been proven to give a good recognition result is :

$l_2(n) = 1 + \frac{L-1}{2} sin(\frac{\Pi n}{L-1}), n = 0, 1, ..., L - 1$

$= 0, else$

Now the cepstrum for the vocal tract, $c_h(n)$, can be calculated by,

$C_h(n)_s(n).l_2(n)$

Mel Scale

The human perception of the frequency contents of sound , for speech signals does not follow a linear scale. Thus for each tone with an actual frequency ,F, measured on a scale called the mel scale .As reference for the mel scale , 1000 Hz is usually said to be 1000 mels. As a nonlinear transformation of the frequency scale, the following formula is used:

$F_{mel} = 2595.log_{10}(1 + \frac{F_{Hz}}{700})$

To apply the mel scale to the cepstrum, a fliterbank of K triangular bandpass filters is applied to $S(z)$.These triangular band pass filters has cepstrum frequencies in K equally spaced mel values. The equally spaced mel values. The equally spaced mel values correspond to different frequency values.

$F_{Hz} = 700.(10^{\frac{F_{Mel}}{2595}} - 1)$

If for example one wants k mel scaled coefficients in the rang 0-5000 Hz(Nyquist range), the first thing to do is to use to get F_{Mel} corresponding to F_{Hz}=5000.Now the calculation of the centrum frequency in the equally spaced mel scale easily be done by dividing the calculated and the reverse operation is done to get back to F_{Hz}.

Now the cepstrum of the triangular filter is found in F_{Hz} and the triangular band-pass filters can be found. The width of each filter is just the distance to the previous cepstrum times 2 .Every triangular filter now will give one new mel spectrum coefficient, M_k, by summing up the filtered result.

Mel-Cepstrum

All steps to create mel-cepstrum coefficients from a speech frame will be described in this section. The sampling rate used in this description is F_s=16 KHz and the block size K=320.The first step is to make the block length to a power of 2 length (N=2^4)$whichenablesafastradix-2algorithmtobeusedforcalculatingtheFFTofblock.InthiscaseK = 320givesanFFToflength512, wh$
$pointFFTto$x$_2(K; m)$ one will get $X_2(n; m)$. After this, the magnitude of $X_2(n; m)$ is calculated and is used with the mel scale filter bank. The mel cepstrum coefficients are then the sum of the filtered result. This can be described by:

$m_k = \sum n = 0^{N-1} |X_2(n; m)| H_k^{mel}(n)$

Where $H_k^{mel}(n)$ is one triangular filter. Here 20 filters are used in the range 0-5000 Hz and these filters are illustrated as FFT vectors.

After the mel spectrum coefficients are calculated , as in the algorithm is taken and the inverse discrete cosine transform is applied as:

$c_s(n; m) = \sum k = 0^{N-1} \alpha_k.log(m_k) cos(\frac{\Pi(2n+1)k}{2N}), n = 0, 1, ..., N-1$

Note that N is now the number of wanted cepstrum coefficients ,not the FFT length. Note also that only K values, m_k , are available for the calculation. Usually N=k, Otherwise one need to zeropad or truncate the m_k values to N wanted values. N values of the frame mel cepstrum are exctracted , $c_s(n; m)$, and choice for removing the pitch is $L = 2/3 * N$.This can be described by :

$c_h(n; m) = c_s(n; m).(1 + \frac{L-1}{2} sin(\frac{\Pi n}{L-1})), n = 0, 1, ..., L-1$

Note that this opration zeros out some of the last mel cepstrum coefficients . After this step the final mel cepstrum values are found. A way to interpret these values usually, is to make an FFT of $c_h(n; m)$ to see its spectral information. The spectral information given is actually $logH(n; m)$, which is the log of the magnitude for the vocal tract of the vocal tract in mel scale . To get the magnitude for the vocal tract filter the following formula can be used :

$|H(n; m)| = |e^{FFT_n c_h(n;m)}|$

Note that the N-points in $H(n; m)$ are now represented in melscale , so $H(n; m)$ is actually in mel scale and not ordinary frequency scale. This is illustrated for a speech file in sampling rate was 16 HZ, the range used for the mel filter bank was 0-5000 Hz and 512 points FFT was used.

3.2 Text to Speech

3.2.1 Extraction of Group of Phonemes

In this part, system detects the group of phoneme and cut them in pieces. In learning process we made table of words, its range for particular group of phonemes and threshold. This table has been tested on various voice samples with same speed of pronunciation. We take the samples of speech from the user according to table.

In the table, for e.g. In word city group of phoneme see/ci is consists 4500samples starting with threshold 0.001. Now while extracting group of phoneme from user training sample we will you this table to detect the exact group. Speed of pronunciation in learning process should be in accordance with the sample audio used for table formation which is mandatory condition for learning process. The table of group of phoneme, range and its threshold is shown below.

3.2.2 Speech Synthesis

In speech synthesis we made the bank containing word and its formation using group of phoneme we extracted in above process.

$Fore.g.Multiplicity = multi+plee+city(multiple- > multi, simply- > plee, city- > city)$

Concatenation is done by using matlab program.

Chapter 4

Summary

Pronunciation corrector is mainly divided into three parts; speech to text conversion and text to speech conversion, Auto-suggestion Graphical Interface. In speech to text conversion there are various methods to extract features from recorded speech from which we decided to implement method using Mel-scale Frequency Cepstrum Analysis. And training and testing will be done using Hidden Markov Model. In text to speech conversion we decided to implement phonetic concatenation method but due to formation gap, concatenation of two phonemes is not giving proper pronunciation, so we decided to use group of phonemes which reduces number of formation gap and gives desired result.

Bibliography

[1] J. W. Picone, ”Signal modelling technique in speech recognition,” Proc. Of the IEEE, vol. 81, no.9, pp. 1215-1247, Sep. 1993.

[2] H. Hermansky, B. A. Hanson, and H. Wakita, ”Perceptually based processing in automatic speech recognition,” Proc. IEEE Int. Conf. on Acoustic, speech, and Signal Processing,” pp. 1971-1974, Apr.1986.

[3] C. Bickley, E. Bruckert, ”Improvements in the voice quality of DECtalk,” Proc. of 2002 IEEE Workshop on Speech Synthesis, Santa Monica, California, pp. 55- 58, Sept. 2002.

[4] K. N. Stevens, ”Toward formant synthesis with articulatory controls,” Proc. of 2002 IEEE Workshop on Speech Synthesis, Santa Monica, California, pp. 67- 72, Sept. 2002.