Non Stationary Noise Removal in Robust Speech Recognition

Monika Sharma[1], Neeru[2]

[1]P.G Student, Department of Electronics and Communication Engineering, ICL IET KUK
[2]Assistant Professor, Department of Electronics & Communication Engineering, ICL IET KUK

Abstract **The evaluation material was Aurora 2.0 that has a noisy speech database, designed to evaluate the performance of ASR systems in these noisy conditions. The source speech datasets used were the standard TIDigits and TIMIT datasets. TIDigits dataset contains digit sequences, whereas TIMIT dataset contains sentences. A recognizer was developed to extract the desired voice from a multi-talker condition. The desired voice was extracted through cepstral subtraction. MFCC feature set was extracted using an average of such l0 log MFB outputs which have a maximum relative variation among them. Then further, the obtained signals are enhanced by performing Dynamic Change Enhancement and Mean Smoothing on the obtained signal.**

Aurora 2.0 dataset consists of 8 different noise scenarios. In our experiments, all the 8 noise scenarios were mixed with the original speech signal one by one at SNR values of 0,5,10,15,20,25,30 dB. The results present the spectrograms of the noise mixed signal at an SNR of 30 dB. The energy of the original signal is compared to the energy of the signal obtained after extraction of the desired signal from the noise mixed signal. Neural Networks were used as classifiers as they have been proved to give comparative results to HMM model. The results obtained by using neural networks for the original dataset i.e. TIDigits has been shown. The results obtained for the TIMIT dataset are 89.26% accurate and those obtained after RASTA filtering are 79.19% accurate.

Keywords: Speech Recognition, Noise Removal

I. INTRODUCTION

A speech recognition system can perform acceptably well in clean environment, however, the performance of the system degrades in a noisy condition. The background noise leads to reduction in the dynamic changes in the spectral energies of the speech signals.The accuracy of the speech recognition systems seems to be high when the input speech is noiseless. However, the accuracy falls down in case of noisy speech. The dynamic changes in the spectra of speech that is given as an input to the recognition system are suppressed due to presence of noise. This reduction in dynamic changes leads to unreliable segmentation of the speech signal, making the task of analyzing the input speech to be more difficult. The traditional method of extracting MFCCs from the speech signal i.e. extracting MFCCs from the log scaled Mel filter banks (MFBs) is susceptible to noise.

The features used for ASR i.e. the log energy and its temporal derivatives appear to be the most important features for speech recognition. They give acceptable results under clean conditions. However, for a low SNR or for noisy conditions, these features appear to be distorted. Therefore, under noisy conditions features derived from full band short time spectral energies and MFCCs produce a mismatch between the training and the testing conditions, resulting in drop in accuracy. This can be proved by the following equations .Let s(i), n(i) and r(i), respectively denote the clean speech, additive noise and resultant noisy speech signals.

$$x(i) = s(i) + n(i)$$

The resultant noisy speech signalx(i) is segmented into I frames using a Hamming Window.

The energy of the noisy speech signal at the I-th frame can be given as:

$$e_x(l) = \sum_{i=1}^{I} x_w^2\,(i) \simeq \sum_{i=1}^{I} s_w^2\,(i) + \sum_{i=1}^{I} n_w^2\,(i)$$

The log energy of the noisy speech signalsx can be computed as:

$$E_x(l) = \log e_x(l) = \log\big(e_x(l) + e_n(l)\big)$$

where

$$e_x(l) = \sum_{i=1}^{I} s_w^2\,(i)$$

and

$$e_n(l) = \sum_{i=1}^{I} n_w^2\,(i)$$

The dynamic changes in the log energy CE can be computed as the difference between the log energies and the present frame l and the subsequent one i.e. l+k. Therefore,

$$C_E = E_x(l + k) - E_x(l)$$

In terms of log,

$$= \log[e_s(l + k) + e_n(l + k)] - \log[e_s(l) + e_n(l)]$$

$$= \log\frac{e_s(l + k) + e_n(l + k)}{e_s(l) + e_n(l)}$$

From the above equation it can be proved that when there is noise in the speech signal, the dynamic changes in the log energy decreases. Therefore, it can be summarized that in the presence of noise, the spectral features like log energy and the MFCCs can no longer represent the variations in the speech signal efficiently. If such signals are input into the recognition system, it will produce a mismatch between the clean speech and the noisy speech or the training and the testing conditions.

Thus, to avoid the above degradation, we find the log energies and MFCCs from the speech signal.

Log Energy Estimation:
Energy is calculated form sub-band spectrum using the log MFB outputs. The reasons behind using the log MFBs are:

- Log MFBs are sub-band based and can capture dynamic variations in the speech signals more efficiently.
- Log MFB outputs with wider change ranges across time can better reflect dynamic variations among speech signals than those with smaller changes.

The relative changes in the log MFB values for the j-th filter bank is given as :

$$R(j) = \frac{X_{max}^{(L)}(j) - X_{N}^{(L)}(j)}{X_{N}^{(L)}(j)}$$

WhereXmax(j) and XN(j) are the maximum values of the j-thlog MFB outputs and the estimated noise log MFB value respectively. XN(j) can be obtained by averaging first several non-speech frames.

II. DYNAMIC CHANGE ENHANCEMENT

The log energy obtained using the above equations still contains noise. So we perform post processing over the signal to further reduce the noise. In Dynamic Change Enhancement (DCE), the noise energy is obtained by averaging the first several non-speech frames of the signal and them subtracting this noise energy from the speech signal.

Thus, the noise subtracted log energy can be given as :

$$u(l) = \begin{cases} E(l) - E_n \text{ ,if } E_l \geq E_n \\ 0, \quad\quad \text{otherwise} \end{cases}$$

DCE is thus implemented as:

$$\breve{E}(l) = \frac{u(l)}{E_{max} - E_n} . E_{max}$$

Where Emax denotes the maximum value of the log energies along the frames in an utterance.

III. MEAN SMOOTHING

The signal obtained after DCE contains high frequency components which contains noise. In order to remove these high frequency spikes from the signal, mean smoothing is performed on the speech signal using a mean smoothing filter. The process of mean smoothing can be explained by the following equation :

$$\hat{E}(l) = \frac{1}{M} \sum_{p=-(M-1)/2}^{(M-1)/2} \breve{E}(l+p)$$

After this process we get log energies and its derivatives after removing a significant level of the noise component from the speech signal. Next we need to find the MFCC coefficients which can describe the actual speech signal reliably.

MFCC Front-End
In the presence of background noise, the log MFB trajectories obtained from the speech signal are suppressed. Thus, these trajectories create a mismatch between the training data and the testing data.Therefore, MFCCs are calculated using the same method as above i.e. by dynamic change enhancement followed by mean smoothing.The dynamic change enhancement is performed as below :

$$\dot{X}^{(L)}(j,l) = \frac{U\left(X^{(L)}(j,l) - X^{(L)}(j)\right)}{X_{max}^{(L)}(j) - X_{N}^{(L)}(j)} . X^{(L)}$$

WhereX(L)(j,l) denote the log MFB output at the j-thfilter bank channel and the I-th frame.$X_{max}^{(L)}(j,l)X_{N}^{(L)}$denote the maximum value in the utterance and the estimated for noise, respectively. U(v) is a unit step:

$$U(v) = \begin{cases} v, \text{if } v \geq 0, \\ 0, \text{othewise} \end{cases}$$

Further, to remove the high frequency noisy spikes from the obtained signal mean smoothing is performed using a mean filter :

$$\ddot{X}^{(L)}(j,l) = \frac{1}{MN} \sum_{(m,n)\epsilon R} \dot{X}^{(L)}(m,n)$$

WhereR denotes the M×N window a $\dot{X}^{(L)}(m,n)$ denotes the neighbors around (j, l).

IV. EXPERIMENTAL SETUP – METHODOLOGY

In Automatic Speech Recognition System (ASR), the project that worked on, the methodology can be divided into four steps as shown in the diagram below.

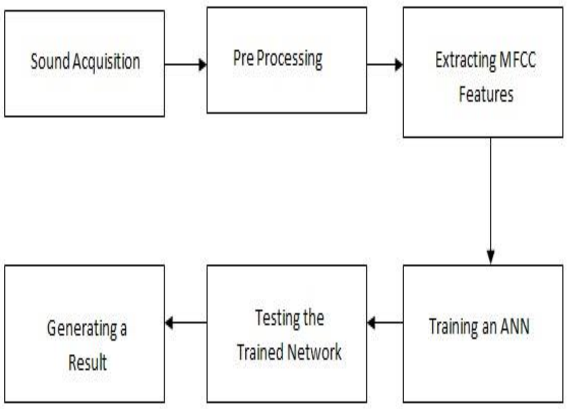

Fig.- General Block Diagram of the Whole procedure in multiple steps as performed

Step 1: Speech Acquisition
The step 1 was the acquisition of sound or speech signal. This work used TIMIT corpus for speech signal. Since this thesis mainly focused on isolated word recognition and not on continuous speech recognition, the speech samples taken were firstly isolated into word sample here and after mentioned as utterances. These utterances were from different speakers both male and female, with different high pass, low pass and band pass filtered applied as well as the original utterance. The utterances were sampled at 16 KHz or 16,000 Hz. The whole speech was segmented into words as “Cottage”, “Cheese”, “With”, “Chives”, “Is”, “Delicious” in the same order as

spoken in the sentence. There were total 6 speakers with 5 speech sample of each speaker as in different leveled filtered samples etc. This gave us a total of 30 speech samples. Thereby, a total of 30 samples of each utterances were extracted out of there 30 samples, i.e. a total of 180 samples were extracted. Out of 30, 20 samples of each utterance were used for training of the artificial neural network and 10 samples were used in testing phase.

Step 2: Pre-Processing

After step 1, the next step was to make a pre-processing block to pre-process each sample before extracting the features.

Pre-processing Step 1:250 Hz High Pass Filtering

This step was done to remove any signal smaller than 250 Hz. This way, low frequency rumbles and 50-60Hz line noise can be removed. Also, a lot cleaner sample is received after this step. To perform this filtering, a 4th order Butterworth filter was used.

Pre-processing Step 2:Pre emphasis filtering

A pre-emphasis filter is used to emphasize the low energy parts like vowels. This filtering enhanced the vowels and thus is a really helpful filter in case of speech signal processing.

Pre-processing Step 3: Spectral noise removal

A spectral noise removal is very important in case of noisy signals. In results section we will see that the recognition rate has reached a very high value just by utilizing a spectral noise removal system.

Pre-processing Step 4: Normalization

Normalization is a very important step, with normalization; we ensure that the sound that is entering the front-end of a recognizer is amplitude invariant.

Step 3: Features Extraction

Step 3 is the extraction of features from the samples. The features we chose for our recognizer are MFCC features along with log Energy output of j-MFBs. This method is based on our base paper on which this work is extended. In the base paper, it is mentioned that if we find take "j" number of log—MFB outputs with these values having maximum relative change parameter, we can achieve a greater accuracy. The first step was to develop a front-end recognizer from the mathematics given in the base paper. The steps can be summarized as below:

1. Using Hamming Window, the sample is divided into four equal parts.
2. Calculating j-MFBs using Maximum Relative Change equation.
3. Calculating 12-MFCC parameters (24 standard MFBs were used here).
4. Using DCE equation to enhance the MFCC coefficient using given equations.
5. Applied a Mean Filter on the 12-MFCC coefficients.
6. Calculated 12 delta parameters (or 1st order differential parameter: rate of change)
7. Calculated 12 delta-delta parameters (2nd order differential parameter: acceleration: rate of rate of change)
8. Calculated log-Energy using j-log-MFB outputs selected in step 1.
9. Calculated its delta and delta-delta parameters.
10. Concatenating them all to make a 4x39 length matrix.

The resultant was a 4x39 matrix with 4 frames and 39 MFCC features. These features were concatenated in a row to form a 1x156 feature vector. This is the final feature vector that is used to train an Artificial Neural Network based recognizer.

Step 4: Training of ANN

Step 4, was to create an artificial neural network for our recognition task. In the base paper, HMM or Hidden Markov Models were used, but Artificial Neural Networks are way better at learning complex correlations in the features, so it was preferred to use ANN for our work.

The ANN that we use is a multilayered feed-forward network. Since we used 156 features as total in a vector, we designed the neural network using 156 nodes in hidden layer and 3 hidden layers having same number of nodes. Also, since we had 6 utterances, we used 6 nodes at the output. Thus our neural network is a 156→156→156→156→156 with bold numbers being input and output layers respectively.

The network was trained using "Scaled Conjugate Gradient Back-propagation" training function. The network was trained using 20 samples of each utterance giving a 120x156 i.e. 120 samples and 156 features each feature matrix. It was trained using supervised learning method, so a 120x6 class matrix was also provided. The class matrix is simply a 0-1 matrix with 0 being false class and a 1 mean true class. This way, for each sample there is only one 1 in the length 6 vector. For example, "Cottage" has class "1" so its label is "1, 0, 0, 0, 0, 0". Similarly, "Cheese" has a class "2" so its label is "0, 1, 0, 0, 0, 0".

V. RESULTS AND DISCUSSION

Once the network was trained, the next step is to perform a testing of the network to see how good it performs. The performance of the network is measured using Confusion Matrices. The whole experiment is designed and tested with MATLAB™ by Mathworks® Inc.

The table above shows all the values of accuracy in percentage (%) for the Original test samples and the samples mixed with different noise scenarios at different SNRs. In the table below, the results with RASTA-MFCC based front-end are shown. The results are clearly deteriorated than with our method

Accuracy in % for TIMIT corpus under different noise scenarios given in Aurora 2.0

SNR	Original	Airport	Babble	Car	Exhibition	Restaurant	Street	Subway	Train	AVG
5 dB		81.7	81.7	85	71.7	81.7	93.3	68.3	88.3	82.97
10 dB		88.3	88.3	88.3	80	80	91.7	80	90	86.84
15 dB		88.3	90	91.7	81.7	86.7	88.3	86.7	91.7	88.90
20 dB		90	91.7	93.3	86.7	88.3	90	90	93.3	90.92
25 dB		95	91.7	93.3	90	93.3	90	91.7	96.7	92.97
30 dB		95	91.7	91.7	91.7	93.3	93.3	91.7	93.3	92.97
0 dB	95%									89.26%

Accuracy in % with RASTA-MFCC Front end processing

Results with RASTA-MFCC front-end										
SNR	Original	Airport	Babble	Car	Exhibition	Restaurant	Street	Subway	Train	AVG
5 dB		70	68.3	73	71.7	73.3	78.3	63.3	78.3	73.3
10 dB		75	76.7	78	71.7	73.3	78.3	73.3	80	76.7
15 dB		81.7	76.7	83	76.7	71.7	83.3	80	80	79.6
20 dB		78.3	80	83	81.7	81.7	78.3	88.3	81.7	81.9
25 dB		81.7	80	83	80	83.3	80	85	85	82.0
30 dB		83.3	83.3	78	78.3	81.7	80	81.7	85	81.7
0 dB	83.30 %									79.19%

In the above results with RASTA-MFCC based front-end, we are getting an average accuracy of 83.3 % with no noise and an overall accuracy of 79.19% with noise scenarios in consideration.

VI. CONCLUSION & FUTURE SCOPE

6.1 Conclusion

In this work, we have successfully implemented an Automatic Speech Recognition Engine using MFCC features and Artificial Neural Networks. The system is tested for robustness via testing under different noise scenarios with different Signal to Noise ratios. The results have been tabulated and analysis has been performed. We have shown that if the base paper technique of using DCE and selecting only j-log MFB Energy parameters for MFCC features are applied with a Feed-Forward Neural Network, the results can be improved by a huge factor. In the base work, only 83% accuracy was attained using HMMs. In our proposed work, we trained a feed forward neural network and attained an average accuracy of more than 95% in case of no noise condition and an overall average of more than 89% with consideration of noise.

6.2 Future Scope

In future, we can extend our work to train the neural networks to recognize phonemes and thus we can create a more accurate and high accuracy phoneme based speech recognition engine.

References

[1] Dr. Shaila D. Apte, " Speech & Audio Processing", Wiley Precise Book, Copyright @2012, Reprint 2013

[2] PreetiSaini, ParneetKaur, "Automatic Speech Recognition: A Review", International Journal of Engineering Trends and Technology, vol.4, issue 2, 2013.

[3] Rashmi CR, "Review of Algorithms and Applications in Speech Recognition System", International journal of Computer Science and Information Technologies, vol 5, 2014.

[4] Ahmad A.M. Abushariah , Teddy S. Gunawan ,Othman O. Khalifa , Mohammad A.M. Abushariah," English Digits Speech Recognition System Based on Hidden Markov Models", International Conference on Computer and Communication Engineering (ICCCE), Kuala Lumpur, Malaysia, 11-13 May, 2010.

[5] B.H. Juang& Lawrence R. Rabiner, "Automatic Speech Recognition- A Brief History of the Technology Development", Elsevier Encyclopedia of Language and Linguistics, 2005.

[6] What is Automatic Speech Recognition? (2009, June), Retrieved may 24, 2015 from

http://www.docsoft.com/Resources/Studies/Whitepapers/whitepaper-ASR.pdf

[7] K.H. Davis, R. Biddulph, S. Balashek, " Automatic Speech Recognition of Spoken Digits", The Journal of Acoustical Society of America, vol.24, no.6, November, 1952.

[8] SadaokiFurui, "50 years of progress in speech and speaker recognition", Acoustical Society of America Journal, vol.116, Issue.4, pp. 2497- 2498, October, 2004.

[9] L. Rabiner and G. Juang, "Fundamentals of Speech Recognition", Prentice-Hall, 1993.

[10] CiniKurian, "A Review of Technological Development of Automatic Speech Recognition", International Journal of Soft Computing and Engineering (IJSCE), vol-4, Issue 4, September, 2014.

[11] Nicholas W.D Evans and John S. Mason, " Computationally Efficient Noise Compensation for Robust Automatic Speech Recognition Assesed Under the Aurora 2/ Frame work", International Conference on Spoken Language Processing, 2002.

[12] Weifing Li, Longbiao Wang, Yicong Zhou, HervBourlard, Qingmin Liao, " Robust Log-Energy Estimation and its Dynamic Change Enhancement for In-Car Speech Recognition", IEEE Transactions on Audio, Speech and Language Processing, vol.21, No.8, August 2013