Phonetic Unification of Multiple Accents for Spanish and Arabic Languages

Saad Tanveer[1], Aslam Muhammad[1],
Martinez-Enriquez A.M.[2], and Escalada-Imaz G.[3]

[1] Department of CS & E,U.E.T., Lahore, Pakistan
saadtanveer@engineer.com, maslam@uet.edu.pk
[2] Departmeent of CS, CINVESTAV-IPN, D.F., Mexico
ammartin@cinvestav.mx
[3] Artificial Intelligence Research Institute, IIIA-CSIC, Spain
gonzalo@iiia.csic.es

Abstract. Languages like Spanish and Arabic are spoken over a large geographic area. The people that speak these languages develop differences in accent, annotation and phonetic delivery. This leads to difficulty in standardization of languages for education and communication (both text and oral). The problem is addressed by phonetic dictionaries to some extent. They provide the correct pronunciation for a word. But, they contribute little to standardize or unify the language for a learner. Our system is to provide unification of different accents and dialects. It creates a standard for learning and communication.

Keywords: Accent unification, Spanish, MFCC, Voice content matching. Pattern matching.

1 Introduction

Four of the most widely spoken languages in the world are Chinese, English, Spanish and Arabic [1]. A large population of human race is speaking these languages, also these people diverse in culture, race and origins due to vast geographical variance. For example, Spanish has many different dialects and some grammatical differences when from geographically different native speakers [2]. These irregularities in the language may lead to miscommunication or even in complete loss of understanding. To come around the problem phonetic dictionaries have been created for Chinese[4] and Arabic [3]. These phonetic dictionaries can correct the pronunciation of a word but do not assist at all in the construction and improvement of the overall accent and dialect. These variations constitute to problem in standardization of language. The same can produce problems in learning the language for oral communication and phonetic correction. To solve this problem of diversity in language, we are interested in development of a system that enables us to:

1. Identify the accent of the speaker.
2. Find the phonetic mistakes of pronunciation and accent. Help to standardize it with phonetic corrections.

J.A. Carrasco-Ochoa et al. (Eds.): MCPR 2012, LNCS 7329, pp. 323–333, 2012.

3. Use identified accent of the person to help narrow down mistakes that particular accent group is prone to make.

Similar problems are addressed by [5] by using voice content matching techniques for Arabic language, the scope is only limited to learning of the Holy Quran. However the system gives a proof that the approach may help create standardization for language.

Every person has unique speaking style that is affected with age, medical condition and primary language of communication. Another reason for the variation is that if the language as a secondary language to speaker. The speaker would carry the knowledge of primary language and make modification when uttering the words in newer language. It creates a problem for accent unification by creating another variation of same language. An accent is defined as "The relative emphasis on syllable or word in punctuation determined by regional or social background of the speaker" [6]. Automated Speech Recognizers (ASR) are also prone to error due to their gender independence. Gender specification independent systems are relatively less accurate when compared to systems that are gender sensitive one by up to 33%[7].

The goal of our system is to identify, the accent of the speaker, deviation from the standard accent and the errors made in utterance on word level. The tool is intended for use with any language but for case study we are emphasizing on Spanish. To achieve this we designed and developed PUMA. It takes a voice stream of a particular sentence from a selected database; the sentence is segmented into voice and silence components. The features of each segment are extracted using Mel-Frequency Cepstral Coefficient, MFCC[10] and a codebook is made using Vector Quantization [8], and Gaussian Mixture models. PUMA identifies its accent by comparing every word uttered with its database of different accents. A mistake if any in utterance is identified along with the nature of error is given as output.

The rest of the paper is as follows. In Section 2 we give the architectural working details of PUMA. In Section 3, related work is given. Section 4 comprises of the experiments and results that our system underwent. Section 5 states the conclusion and future work.

2 PUMA Architecture

PUMA uses MFCC for extraction of features of voice streams to generate multi classifier code books of each input stream. PUMA uses MATLAB framework using signal processing toolkit. The voice input of the user is processed by undergoing data processing that filters background noise and trims the voice sample of silence at beginning at end of the stream. It then moves to the word slicing module that removes the silence between the words and stores them separately. As with any ASR utility, PUMA also contains training and testing phrases as shown in Fig.1. In the training phrase, the standard speech databases of different accents are created. In order to improve the recognition rate of the system we created separate databases on basis of gender. As the pitch and phonetically diverse voices of men, women and especially children lead to wrong results.

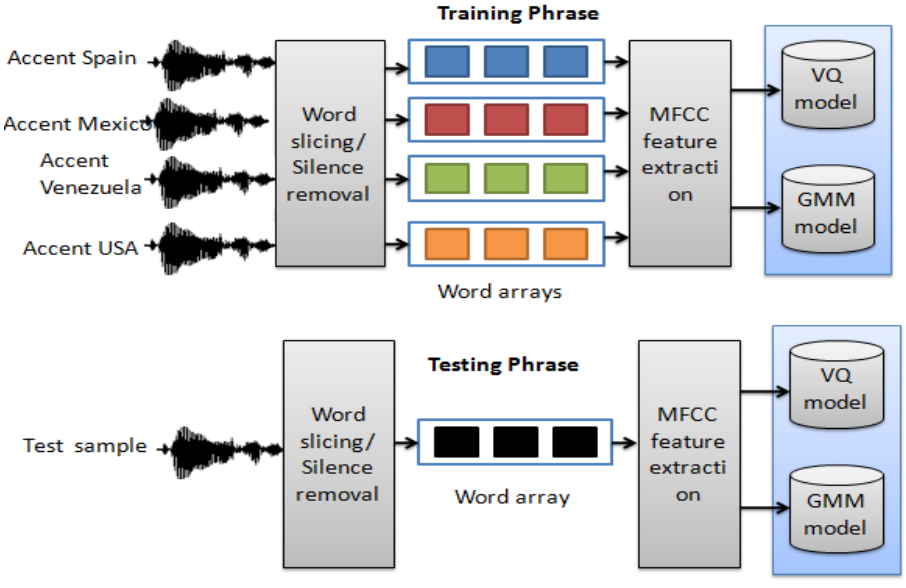

Fig. 1. Architectural modules of PUMA

In testing phase, the stream of voice undergoes similar process for noise filtration and feature extraction and the model are stored for the test sample. Gaussian Mixture Models and Vector quantization models are used to identify the accent of the speaker and identification of pronunciation/utterance related mistakes respectively.

Our proposed system does not only improve the work done by [5] but also provides a tool that is not only limited for learning of Arabic or Spanish; it can be expanded to other languages. Mistakes that are to be detected and reported by PUMA:
1) Missing words in recitation, 2) Inappropriate sequence of words in utterance, 3) Identification of which word has been identified wrong, 4) Switching to dissimilar words, and 5) Handle repeating words.

All the above stated problems are detected and communicated to the user for correction so that more surgical corrections can be made saving time and effort in learning. The score of matching of individual words would enable the system to identify the accent origin of the user using GMM. The detail of the working modules of PUMA is as follows.

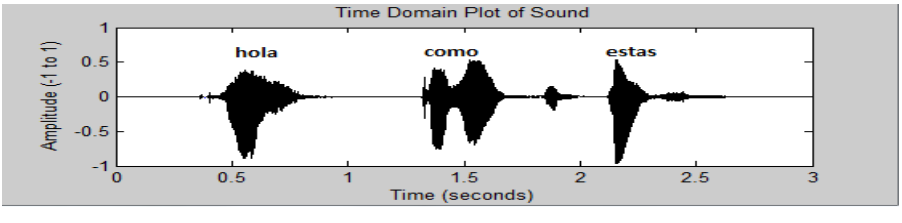

Fig. 2. The Time domain plot of a phrase in Spanish with silence between words

2.1 Data Preprocessing

The input stream is recorded on an 11025 Hz, 16-bit, 1–channel mode as Fig. 2. This stage of process is of prime importance for getting results with lower error probability. Background noise and frequency ranges(outside human communication range) are removed. To get better results, we need to have content rich voice stream, with high Signal to noise ratio(SNR), this is achieved:

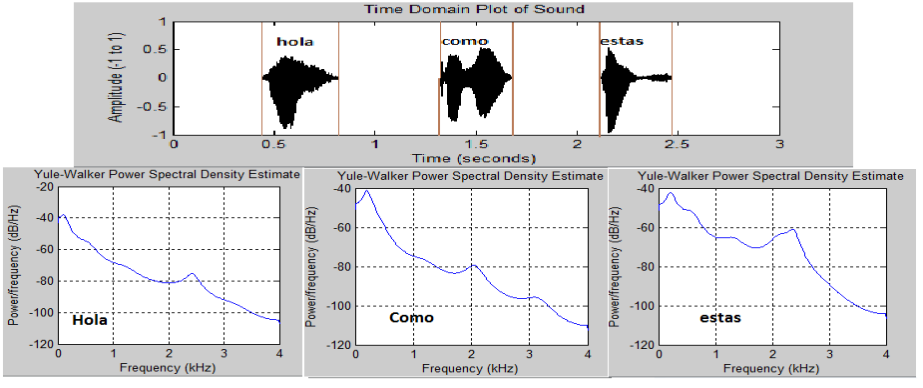

Fig. 3. The acoustic model of the verse, after words are extracted (top). Each word is then stored separately with Power spectral density estimates of words (bottom).

- **Signal Emphasis and Background Noise Filtering.** Unwanted frequencies are nullified and voice content signal in amplified by giving raise to higher magnitude frequencies with respect to lower, in order to improve the SNR. Noise cancellation is applied to remove echoes in the signal if any. First order Finite Impulse Response (FIR) filter is applied on digitized signal to achieve this. The FIR filter is calculated by Eq. 1. To get a gain of 18db, we selected α=0.935 for pre-emphasis parameter, which may have value close to 1. It removes the white noise, and the inter-modulation noise to improve the SNR. The filter is designed to isolate the frequencies with possible human voice content and remove all other components giving a content rich stream. As Fig 3(bottom) also gives evidence that most of the speech content in human voice in largely contained in the range less than 4000Hz.

$$H(z) = 1 - \alpha z^{-1} \tag{1}$$

- **Word Extraction and Array Management.** This module extracts the words in audio stream, so the system can store all words uttered separately into the database. This is done to ensure that we can match the error on word level as the comparison done is not on the whole audio stream but on the words that have been uttered. The detection of words is performed using short-term energy method [9] with combination of zero crossing. The voice speech is separated from unvoiced speech on basis of two metrics, i.e. threshold energy and zero crossings. Zero crossings rates are higher in unvoiced speech so they are filtered out. Whenever there is a sequence of frames having consecutive frames with energies higher than the threshold and lower zero

crossing rates, the segment of stream is saved as an entity in the array. Using both of these techniques for words separation there is lower probability of losing useful information during word slicing. This process continues till the end of stream and finally we get all the words in the stream (see Fig 3 Top).

2.2 Feature Extraction Using MFCC

Mel-Frequency Cepstral Coefficient (MFCC) transformation for feature extraction exploits auditory principles of voice and also the decorrelating property of cepstrum. Another interesting property of Mel cepstral is that is amenable to compensation for convolutional channel distortion. It is the most successful feature representation method for feature representation in speech related tasks, because it simulates the behavior of human ear and uses Mel Frequency scale using 12 filters mel filter bank for our system is shown in Fig .4. The MFCC feature extraction technique consists of seven major components which are:

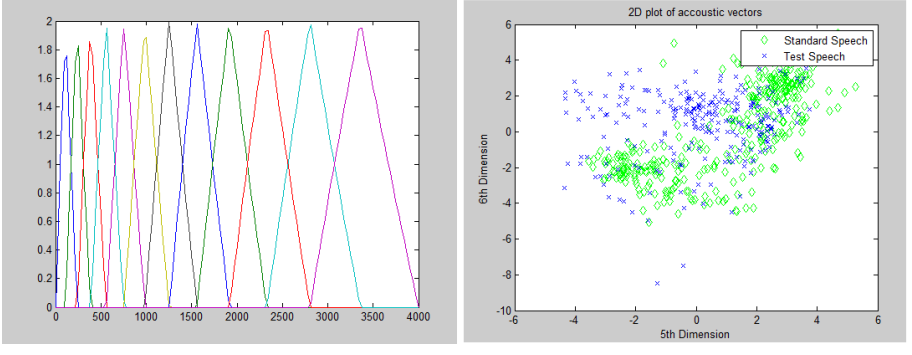

Fig. 4. Block diagram of MFCC Filter bank using 12 filters(left) and Process of Matching VQ codebook of words(right)

Framing and Windowing. In order to perform any useful signal processing on a signal, it is imperative it undergoes Fast Fourier Transform (FFT), short intervals of speech signal to be used for FFT as its is prerequisite for FFT calculation. The segment of speech is 24 mS. In order to prevent data loss, we further create blocks by overlapping half of each frame with next one so that no information is lost i.e., each frame would contain 12 mS of previous frame. Hamming window is used of same size as that of the frame. This resultant frame has zero energy at the start and at the end, simplifying calculation by removing signals from frame boundary. The Hamming window is obtained by Eq. 2:

$$w_n = 0.54 - 0.46\cos\left(2\pi n / N-1\right) \qquad \text{where } 0 \le n \le N-1 \qquad (2)$$

In Eq. 2, N is the total number of samples in each frame in our case it is 256 and n is any value from range 0 to N-1. The value of N is chosen after careful optimization between the number of samples and the number of frequency features that can be included in each of the window.

Discrete Fourier Transformation(DFT). In order to transfer each windowed frame from time domain to frequency domain, Discrete Fourier Transformation is applied by use of FFT algorithm. The windowed signal is input to DFT and the output of this is a complex number, representing each frequency band (0 to N-1) having magnitude and phase of that frequency component in original signal. The DFT is obtained by Eq. 3:

$$Y_{fft}[n] = \sum_{k=0}^{N-1} Y_{windowed}[k]\, e^{-2\pi j k n / N} \tag{3}$$

Where k= 0, 1, 2, 3, …, N-1 and Y2[n] is the Fourier Transform of Y1[k].

Mel Filter Bank. The frequency energy distribution graphs in Fig. 3 show that low frequencies in speech signal contain more useful information as compared to higher ones. To emphasize on these low frequency components, Mel scale is applied. The formula used to calculate Mels for a frequency *f* in Hz is given in Eq. 4:

$$\text{Frequency} \quad (\text{Mel Scale}) = 2595 * \log 10 \ (1 + \frac{f}{700}) \tag{4}$$

Logarithm and Inverse Discrete Fourier Transformation. Humans are prone to change their voice levels while speaking, so the system has to be less sensitive towards small changes in voice levels for this reason Log is applied to the signal. The behavior of human ear is logarithmic; log mimics this effect in the signal. IDFT converts the speech signal back to time domain from frequency domain. MFCC extracts feature vectors from input signal(Eq. 5) :

$$Y\_cep[k] = \sum_{n=0}^{N-1} x[n] \cos\left[k\left(n - \frac{1}{2}\right)\frac{\pi}{N}\right], \qquad k = 1,2,3,..., L \tag{5}$$

Here x[k] is the logged value of each Mel filtered speech segment gained from previous step. L is the required number of Mel Cepstral Coefficient taken from N filter tapes of each frame and in our case L is 12.

2.3 Modeling, Storing and Comparison of Codebooks

For storing the models of speech that are generated from MFCC, we used two methods vector quantization (VQ) and Gaussian Mixture models (GMM). For optimal modeling, only 12 Mel coefficients were used to model the voice models. The numbers of features vectors are reduced by getting highly representative vectors which is achieved by VQ technique[8] and GMM.

GMM[13] is independent temporal order of the signal. The probability of identification for a positive match is given by the formula given in Eq. 6:

$$\left[\text{Pgmm}(\ V_k / \lambda) = \ = \sum_{i=1}^{N} w_i \cdot y_i (v_k) \right] \tag{6}$$

Where V_k are the component mixture densities and w_i are the mixture weights. The voice model for an accent is shown in Eq. 7:

$$\lambda_j = W_i \cdot \mu_i \cdot \sum i \tag{7}$$

In the case I is the index of mixture. The likelihood of a match has been made or not is given by the relation given by Eq. 8:

$$L_j = \sum_{f=1}^{F} \log p(v_k / \lambda) \tag{8}$$

Here F is the number of frames in the voice stream and the MFCC index is represented by f. The relative independence of GMM from time variance makes it and excellent choice for matching the accent with added text independence. GMM results likelihood matches with different accent for each word with all four possible matches for accent groups

Vector Quantization maps the vocal tract function by using only two poles, which implies that there are only four reflection variables enabling us to compress the speech using 8 bits per frame to code the function of vocal tract. In the case of VQ, we investigate the distortion between two voice samples. The power density function is plotted for VQ models and the centroids are formed for each sample given by Eq. 9:

$$\hat{x} = VQ [\underline{x}] = \underline{c}_i \text{ for } \underline{x} \in C_i \tag{9}$$

The vector $\hat{x}$ is the vector chosen from M possible quantization VQ is the Vector quantization operator, and c_i is the possible reconstruction levels.C_i is the cell boundary. The vectors are then plotted and similar vector points are clustered. Mean is calculated for all features vector. So, to calculate the mean of a set of K vectors, Eq. 10 is used:

$$M_j = \frac{\sum_{i=1}^{k} x_i}{k}, \qquad j = 1,2,3,...,12 \tag{10}$$

In order to generate clusters of vectors, the distance between each feature vector and Means values is calculated through Euclidean Distance Formula. Fig .4(right) gives the plot of the vectors of a words segment from test speech being compared to respective word in standard speech of an expert. The array of words processed one by one during training stage and the input taken during testing phrase is also compared using the Comparison module.

2.4 Comparison Module

This module is responsible for finding the mistakes in sentences and accent identification this is achieved as: Identification of accent is done by matching each input word, with the database of codebooks of GMM, highest commutative average of words that matches with a particular accent, is the accent of the speaker. If PUMA detects the similarity probability of GMMs with the standard accent lower than the acceptable threshold, it is reported to the speaker to try again.

The words that are missed in utterance are identified by distortion metric using VQ codebooks. The result array is analyzed; if some word(s) is missing from the sequences, the missing word is identified on the interface of PUMA. Speakers are

prone to utter phrases that sound similar. If the uttered words match similar words in the database of codebooks the user is notified to rectify the mistake and try again. Humans are also prone to repeat a part of sentence when trying to put together words and people often repeat parts of sentences, PUMA can identify the repeating words and ignores the redundancy to make sure that results are not altered. These are identified to user from the result array mismatches also the user is notified the phrase they switched to. Another possible mistake is that users may utter the words in wrong sequence. The system is intelligent to identify if the words uttered are correct phonetically but are in wrong sequence.

In case of switching to entirely different phrase, if these words are in the database of the system user is notified for the words that were closest match of words uttered. The matching module is shown in Fig. 5 that shows matching user input word array with possible matches. To ensure ability to identify words, it is compared with multiple streams of experts with correct utterance. The average of all matches of experts of that gender group is displayed to user by the system.

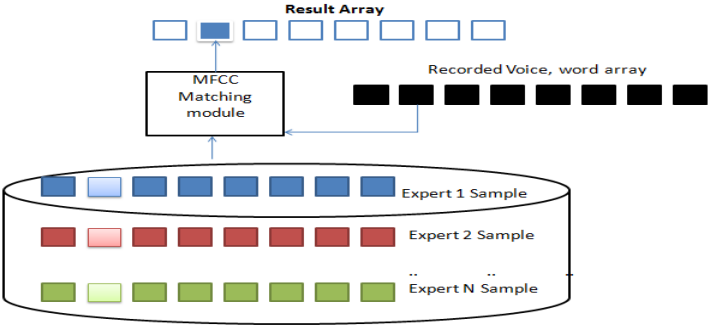

Fig. 5. Comparison and result Matching module

3 Related Work

There have been many efforts to detect the accent like [11] suggest that accent aware speech recognition has lower ratio of error that accent independent speech. They implemented Phoneme–class HMM. Gender and accent information can also help optimize the process of speech recognition [12].However it does not provide any assistance in identification of mistakes in pronunciation.

Similar approaches have been adopted in [14] using GMM for accent identification. But the scope of study was limited only to American and Indian English The idea of accent specific speech recognizer is presented in [12].

Phonetic dictionaries are generally made from only a specific accent group limiting their diversity to choose a standard. This led us to make database of standardized speech from different accent origins to improve the error detection ability of our system. An attempt to correct the pronunciation was presented in [5], where MFCC was employed to find mistake in voice streams. These streams were matched against a standard voice sample and the difference between them is given. Their system

however lacked the ability to pin point the mistake in the sentence. The problem of accent standardization can not only be solved without proper training of pronunciation. PUMA provides one stop solution for both of these problems i.e. phonetic corrections with accent identification.

4 Experiments and Results

To verify our findings we took a case scenario of Spanish. For our study, we included people with different 4 accents: 5 men, 5 women, and 5 children were selected. We took 15 different sentences and asked experts to utter those sentences. PUMA is designed to work not only for Spanish or English it can be used for any languages or accents. Any accent or language group can be selected as Standard criteria. In our case we chose streams from native Spanish speakers. Few of the examples of sentences are as follows: 1) uno dos tres cuatro cinco seis siete ocho nueve diez, 2) este es un día hermoso, and 3) hola como estas.

Table 1. PUMA Results For Accents

Accent Origin	Correct Accent Registration percentage
Spain	82 %
Venezuela	90%
Mexico	87%
United States of America	94%

The ability of PUMA to detect error, was tested by taking 10 test subjects of each gender group and the results that are shown in Table 1 is the average score of the performance of each gender category. The accent matching was achieved through GMMs. Identification of accent is done by matching each word that is input with the database of codebooks, highest commutative average of words that matches with a particular accent. The second phase of testing was designed for testing against mistakes of pronunciation based on the VQ models of words with results in Table 2.

The results from both the tables show promising results for both accent recognition and also in identification of mistakes in sentence utterance. All categories get satisfactory registration with only one exception. Words that are phonetically similar can lead to false positive hits in some cases, due to similarity in model of the words that are phonetically similar.

Table 2. PUMA Fault Detection Results

Type of Error	Percentage of Fault detection
Incorrect Pronunciation of words	81 %
Missing words	95%
Incorrect sequence of words	98%
Switching towards similar words	69%
Switching to Dissimilar words	87%

5 Conclusion

We claim that Phonetic unification can be performed by using MFCC to extract features from voice stream from segmented sentences, use GMM for accent identification and VQ models for Phonetic error detection. The case of Spanish language with four accent groups is just one implementation of PUMA. It can be expanded to other languages, enabling us to solve standardization problems and also identify the accent of the speaker. Also the resolution of error detection has been improved making the system intelligent to detect word level mistakes. Other mistakes like missing words, sequences etc. are also handled. For future prospect similar implementations can be made for English and other languages too. PUMA can be a formidable tool for learning and standardization of accent for any language also capable of identifying other phonetic mistakes. To make the system more versatile the ability to find sub-word mistake in streams can be integrated into the system. This tool can make an immense contribution to under resourced languages too for which the availability of experts is a problem.

References

1. ISSN 1553-8133, https://www.cia.gov/library/publications/the-world-factbook/fields/2098.html?CountryName=&countryCode=&regionCode=o
2. http://en.wikipedia.org/wiki/Spanish_dialects_and_varieties
3. Ali, M., Elshafei, M., Al-Ghamdi, M., Al-Muhtaseb, H., Al-Najjar, A.: Generation of Arabic Phonetic Dictionaries for Speech Recognition. In: International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates, p. 59 (2008)
4. Zhou, W., Yuan, B., Miao, Z., Tang, X.: Error correction via phonetic similarity-based processing for Chinese spoken dialogue system. In: Proceedings of the 8th International Conference on Signal Processing (ICSP), Beijing, China (2006)
5. Muhammad, W.M., Muhammad, R., Muhammad, A., Martinez-Enriquez, A.M.: Voice Content Matching System for Quran Readers. In: Proceedings of ninth Mexican International Conference on Artificial Intelligence, MICAI, Pachuca, Mexico, pp. 148–153 (2010)
6. Malhotra, K., Khosla, A.: Automatic Identification Of gender and Accent in spoken Hindi Utterances with regional Indian accents. In: Proceeding of Spoken Language Technology Workshop, SLT IEEE 2008, Goa, India, pp. 309–312 (2008)
7. Schultz, T., Waibel, A.: Language Independent and Language adaptive acoustic modeling for speech recognition. Speech Communication 35(1-2), 31–51 (2001)
8. Gray, R.M.: Vector Quantization. IEEE ASSP Magazine, 4–29 (1984)
9. Greenwood, M., Kinghorn, A.: SUVing: Automatic Silence/Unvoiced/Voiced Classification of Speech. Undergraduate Coursework, Department of Computer Science, The University of Sheffield, UK (1999)
10. Ibrahim, N.J., Razak, Z., Yakub, M., Yusoff, Z.M., Idris, M.Y.I., Tamil, E.M.: Quranic verse Recitation feature extraction using Mel- Frequency Cepstral Coefficients (MFCC). In: Proc. of the 4th IEEE Int. Colloquium on Signal Processing and its Application (CSPA), Kuala Lumpur, Malaysia (2008)

11. Kat, L.W.: Fung, P.: Fast accent identification and accented speech recognition. In: 1999 IEEE Proceedings International Conference on Acoustics, Speech, and Signal Processing, vol. 1 (1999)
12. Deshpande, S., Chikkerur, S., Govindaraju, V.: Accent Classification in Speech. In: Proceeding of the Fourth IEEE Workshop on Automatic Identification Advanced Technologies, pp. 139–143 (2005)
13. Quatier, T.E.: Discrete time speech processing principles and practice (2004)
14. Chen, T., Huang, C., Chang, E., Wang, J.: Automatic Accent Identification Using Gausssian Mixture Model. In: Automatic Speech Recognition and Understanding, ASRU 2001, Madonna di Campiglio, Italy, p. 343 (2001)