Electronic Journal «Technical Acoustics»
http://www.ejta.org

2008, 20

M. Chandrasekar[1], M. Ponnavaikko[2]

[1]*RVSCET, Coimbatore, India, e-mail: chandramu2000@yahoo.co.in*
[2]*Bharathidasan University, Tamilnadu, India, e-mail: ponnav@gmail.com*

Tamil speech recognition: a complete model

Received 12.11.2008, published 11.12.2008

The paper presents a new method to build a speech recognition system to recognize spoken Tamil. Continuous speech recognition in Tamil is a long desire of the researchers in Tamil computing. This paper presents an algorithm developed for segmenting the speech signal and then a segment-based recognition system. The new approach segments the words from the speech and then characters from words. Then back propagation algorithm is used to train and identify the segmented characters. The proposed method is used to train and test the speech of a given individual. It can be extended to work independent of individuals. The system developed is found to be efficient and effective.

INTRODUCTION

Research in automatic speech recognition (ASR), aims to develop methods and techniques that enable computer systems to accept speech input and to print the output as text on the screen. Speech recognition research for Tamil language is at the early stages of development. A survey of the research contributions towards speech processing of Indian languages made by the authors is given in [1]. Considerable amount of research work has already been carried out for recognizing English speech [2–64]. There is an urgent need for the recognition of the speech in regional languages.

Suzuki and Nakata built a special hardware system to recognize Japanese vowels having an elaborate filter bank analyzer [65]. The digit recognizing hardware of Nagata from Japan paved the way for highly productive research in speech recognition for Japanese language [66]. Jean-Marc Boite and Christophe Ris built a phone recognition experiment with baseline system which achieved a phone accuracy of about 75%. A French speech recognizer was developed using hybrid Hidden Markov Model(HMM)/Artificial Neural Networks(ANN) [67]. Solomon Teferra Abate and Wolfgang Menzel developed a syllable based Amharic speech recognition system using Hidden Markov Modeling, and achieved 90.43% word recognition accuracy [68].

Chandrasekar proposed an approach to recognize Consonant-Vowel (CV) units in Indian languages using artificial neural network [69]. Gangashetty et al. presented neural network models for recognition of syllable like units in Indian languages [70]. Auto Associative neural network (AANN) is used for non-linear compression of feature vectors. Multilayer Feed Forward neural network (MLFFNN) is used for preliminary classification of 145 CV units into 9 subgroups. Nayeemulla Khan developed speech database for Tamil with 30688 words

and Telugu languages with 25463 words [71]. Prasanna proposed a method to find begin and end points of speech based on Vowel Onset Points (VOP) [72]. S.V.Gangashetty et al presented a method in which auto associative neural network is used to get the features of CV utterances [73].

A large annotated Hindi speech database was developed [74] for training a speech recognition system for Hindi. This paper describes about segmentation and labeling in terms of acoustic phonetic units and prosodic characteristics of Hindi vowels. Continuous speech recognition system for Hindi [75] recognizes spoken sentences for railway reservation system. Mel frequency cepstrum coefficients (MFCC) are calculated and used as feature vector. Hidden Markov Model is used to characterize the temporal aspect of speech signals. Continuous speech recognition system for Hindi [76] describes various improvements carried out to the speech input system and the resultant increase in performance. The sentence accuracy is improved by using delta MFCC. Cepstrum mean normalization is employed to minimize the variations due to speaker, ambient acoustics, microphone, transmission channel etc. Speech is collected from more members for training so that the performance of the system is improved further. Durational characteristics of Hindi phonemes in speech were dealt by Samudravijaya [77]. Continuously spoken sentences from an annotated time aligned database is used in this experiment. The means and durations of various units of consonants and vowels are computed. The interesting behavior of decrease in duration of segment with an increase in duration of acoustic segment is reminiscent of a general trend observed in pairs. Chalapathi Neti developed a continuous speech recognition system for Hindi [78] in which 64 phonemes of Hindi phone set was used. A trigram language model is built so that the probabilities are distributed over all sentences. Large amount of speech database is used to increase the accuracy of the system.

Paul Mermelstein [79] described a new segmentation algorithm which allows assessment of the significance of loudness minimum which will be a potential syllabic boundary from the difference between the convex hull of the loudness function and the loudness function itself. Andre G. Adami and Hynek Hermansky proposed a method to segment sentences into a sequence of discrete units, such as phonemes [80]. Ramil G. Sagam et al. used Multi Layer Perceptron technique to segment sentences into words, syllables and phonemes [81]. A. Lipeika et al. described about segmentation of words corresponding to the phonetic events [82].

This research work aims to develop a system for recognizing the speech in Tamil. Tamil language is structurally different from English and other languages in the world. Even among the Indian languages, Tamil is different structurally and phonetically. In Tamil, the pronunciation of independent letters and a group of letters forming words are not different. The Tamil speech recognizing system does not require the support of a dictionary. Thus the recognizing process in Tamil speech is fairly simple compared to English. This paper presents a complete model to recognize the Tamil speech spoken by a given individual.

1. METHODOLOGY

The system developed by the authors for recognizing Tamil speech has the following three stages.

(i) Segmentation of words from the speech signal.
(ii) Segmentation of characters from the segmented words.
(iii) Recognition of the characters.

The methodology proposed does not require the support of dictionaries. The speech is segmented into words and characters. Software is written for segmentation and incorporated in Matlab.

1.1. Segmentation of words

The recorded wave of speech signal over a time scale is used for distinguishing words from the spoken sentences. It is difficult to segment the sentences from the spoken speech signal since the punctuations used are not known in the speech signal. The words are separated by silence states in the recorded signal. The silence state may be short or long. Hence, by sensing the short or long silence state the words are segmented.

The utterance of a single speaker is recorded and stored. This speech signal is given as input to the software developed in Matlab and the energy contour of the speech signal is obtained.

The energy of speech signal [83] is given by E_n in equation (1).

$$E_n = \sum_{m=0}^{N-1} [w(m)x(n-m)]^2 , \tag{1}$$

where $w(m)$ is the weighting sequence or window which selects a segment of the discrete-time signal $x(n)$, and N is the number of samples in the window.

Figure 1 shows the energy contour of a typical speech signal **அம்மா இங்கே வாவா** over time.

The function E indicates time varying magnitude properties of the speech signal. Energy function provides a good measure for separating voiced speech segments from the silence state between speech segments.

For clear identification of the silence states in the energy value graph, the x-axis is shifted by a value of u. After careful analysis of many sentences a trial value of 0.2 is found to be optimum for u.

The graph of the energy signal is sampled using Matlab software. The energy values are stored in a text file. Energy graph of the speech is continuous as shown in figure 1.

The energy values from the text file are processed and the words are segmented from speech signal by using the following algorithm.

- For the example considered, if the energy value of the sample is less than or equal to u, the energy value of the sample is set to be equal to 0.0. With this, the zero cross over points are obtained for locating starting and end points for the word. The resultant wave for the example will be as shown in figure 2. In figure 2, the segments 0-1, 2-3, and 4-5 are the silence states and the segments 1-2, 3-4 and 5-6 represent the words **அம்மா இங்கே வாவா**.

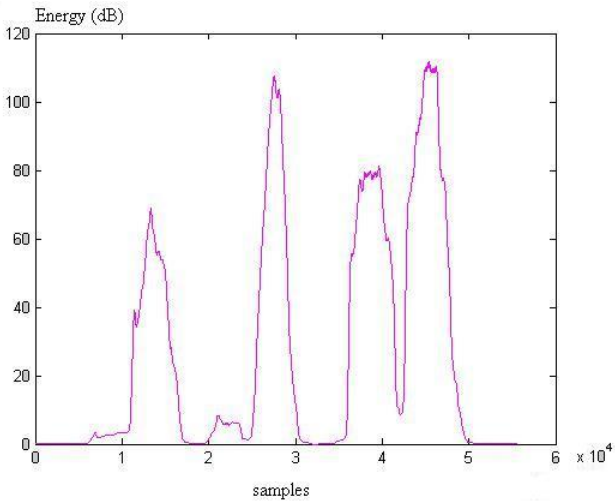

Fig. 1.

Energy value of the graph of the speech: **அம்மா இங்கே வாவா**

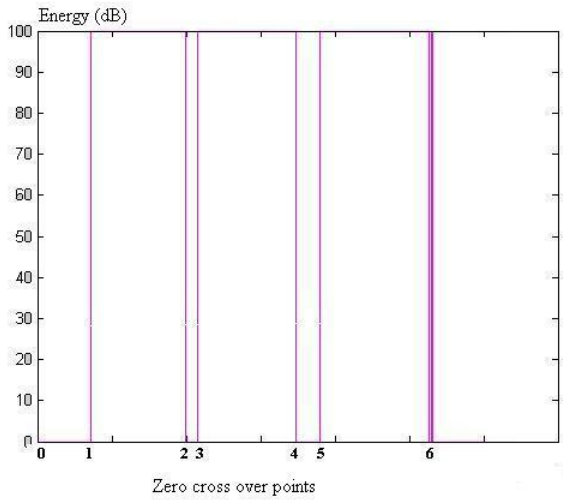

Fig. 2.

Resultant wave of example sentence

The segmented words are stored as separate waves. The wave form of a segmented word "**அம்மா**" is shown in figure 3.

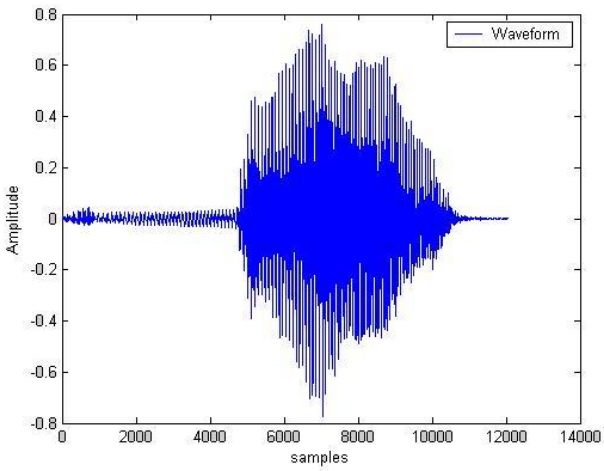

Fig. 3.

Wave form of the word "**அம்மா**"

1.2. Segmentation of characters from the segmented word

Using the algorithm discussed in the section 1.1., the words are segmented from sentences and stored as sound waves. This stored speech signal is considered further for segmentation of characters from the word. The characters are represented by a rapid rise and fall in signal energy graph. The rate of change of energy is expected to be large at the start and at the end of the character. However there may be rapid changes in energy in the middle of the region or at any point of the segment of wave. A new method for detecting the boundary of the character is proposed.

Algorithm for segmentation of characters from the speech signal shown in figure 3 is given below.

1. The segmented words from the speech signal is given as input to the software developed using coding in Matlab for segmenting the characters and the energy contour of the segmented word signal is obtained (figure 4). Store the energy values of the wave in a text file. The energy curve of the word is divided into different segments with each segment having ten samples. For each segment, 10th sample is considered and stored in another text file.
2. The speech energy graph is smoothened by using moving average method and the resultant values are stored in a text file.
3. Determine the turning points at which the energy value is increasing or decreasing.
4. Group the closely associated values sequentially and select the least value in each group removing the other values and store.
5. Check the relativity of the values from the second point onwards and determine the turning points. Group the segments. Retain the first value of each group and remove the rest. Store the resulting values.

This results in segmentation of word into characters. The starting and ending values of the segments corresponding to the required character is stored in to a text file for further processing.

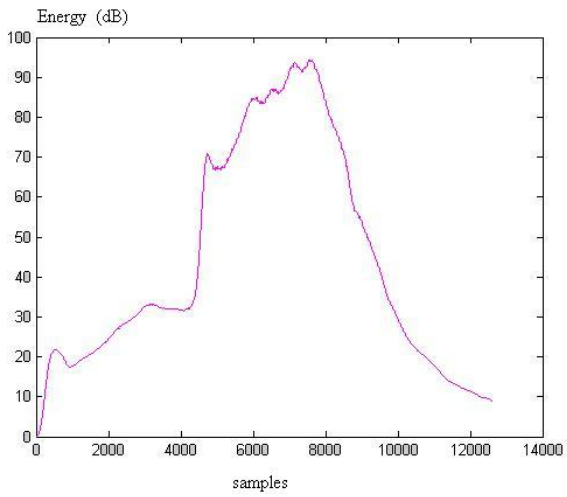

Fig. 4.

For the speech signal: “அம்மா ”

1.3. Recognition of characters

1.3.1. Characters to be trained

The Tamil language contains vowels, consonants and vowel consonants. While speaking the vowels and vowel-consonants when not followed by a consonant, sounds phonetically as independent characters. But, when a consonant follows a vowel or vowel-consonant, the vowel and the following consonant as well as the vowel consonant and the following consonant, sounds like a single phoneme. Hence in the recognition process, these combinations, vowels followed by a consonant and vowel-consonants followed by a consonant are treated as single combined characters. Thus, the number of characters to be considered for training includes 247 stand-alone characters and over 900 combined characters.

To have a complete recognition system, all the stand-alone characters and combined characters are to be trained. For each character/combined character, 20 patterns are given as input for training. One more pattern for each character/combined character is considered as a standard pattern. These characters are trained by using the training algorithm discussed in section *1.3.2*.

1.3.2. Training Algorithm

The scheme proposed by the authors to recognize characters, uses Back Propagation Network (BPN) paradigm of neural network with the cepstral coefficients obtained with the help of Matlab software. For each character, a selected number of sample training waves and target wave of the speech is considered for training the system. The normalized cepstral coefficients of training waves and target wave are presented as input to the input layer of the BPN [52].

The BPN algorithm for training the system is as given below.

1) Present 18 set of cepstral coefficients of the first training wave and 12 cepstral coefficients of the corresponding target wave T_k as input to the input layer.(18 input neurons and 12 output neurons).
2) Obtain the sum of the weighted inputs to the hidden layer y_j using (2).

$$y_j = \sum w_{ji} x_i , \tag{2}$$

where x_i is the input cepstral coefficients, which connects the jth unit of the hidden layer and w_{ji} is the weight associated with that connection; i = 1 to n, j = 1 to h where n is the number of neurons in the input layer and h is the number of neurons in the hidden layer.(15 hidden neurons)

3) Generate the output of the hidden layer z_j, transferring y_j using a non-linear activation function $g(.)$.

$$g(y_j) = \left(\frac{1}{1 + exp(-y_j)} \right), \tag{3}$$

$$z_j = g(y_j) \tag{4}$$

$$= \left(\frac{1}{1 + exp(-y_j)} \right). \tag{5}$$

4) Obtain the sum of the weighted inputs to the output layer a_k using (6).

$$a_k = \sum w_{kj} z_j , \tag{6}$$

where z_j sends connection to the kth unit of the output layer and w_{kj} is the weight associated with that connection; k = 1 to m where m is the number of neurons in the output layer.

5) Generate the final output of the output layer O_k, transferring a_k using a non-linear activation function $g(.)$.

$$g(a_k) = \left(\frac{1}{1 + exp(-a_k)} \right), \tag{7}$$

$$O_k = g(a_k) \tag{8}$$

$$= \left(\frac{1}{1 + exp(-a_k)} \right). \tag{9}$$

6) Compare output activations O_k, to the target values T_k for the pattern and calculate error E for the given pattern using (10).

$$E = \frac{1}{2} \sum (T_{k-} O_k)^2 . \tag{10}$$

7) Now to propagate the error backwards, calculate its error signal δ_k of the output layer neurons using (11).

$$\delta_k = O_k (1 - O_k)(T_{k-} O_k) . \tag{11}$$

8) Then calculate the error signal δ_j of the hidden layer neurons, using (12).

$$\delta_j = z_j (1 - z_j) \sum w_{kj} \delta_k . \tag{12}$$

9) Calculate the delta weight Δw_{kj} and update the network weights w_{kj} between hidden to output layer using (13) and (14).

$$\Delta w_{kj} = \eta \delta_k z_{kj} + \alpha \Delta w_{kj} , \tag{13}$$

$$w_{kj}(new) = w_{kj}(old) + \Delta w_{kj}\,, \tag{14}$$

where η is the learning rate, z_{kj} is the input from unit *j* into unit *k*, and α is the momentum term.

10) Calculate the delta weight Δw_{ji} and update the network weights w_{ji} between input to hidden layer using (15) and (16).

$$\Delta w_{ji} = \eta \delta_{\mathrm{j}} x_i + \alpha \Delta w_{ji}\,, \tag{15}$$

$$w_{ji}(new) = w_{ji}(old) + \Delta w_{ji}\,. \tag{16}$$

11) Present 18 cepstral coefficients of the second training wave and 12 cepstral coefficients of the target wave as input to the input layer.
12) Repeat the steps 1 to 11 with the updated new weights until all the 20 training patterns are presented.
13) Sum the errors of all the training patterns, and obtain the average of the errors and compare the same with the specified tolerance.
14) Repeat the entire process from steps 1 to 13 till the convergence criteria is met.
15) Repeat steps 1 to 14 to train the other characters.

The optimized weights are the output of the training program. These are stored in a text file.

2. APPLICATION

Consider the spoken sentence **அம்மா இங்கே வாவா**. The energy graph of the wave signal is plotted using Matlab software. The energy values are stored in energy1.txt and the graph is shown in figure 1.

The words of this speech signal are segmented following the procedure discussed in section 1.1. The value for *u* is set at 0.200. The segmented speech signals of the three words are shown in Figures 4, 5 and 6 respectively.

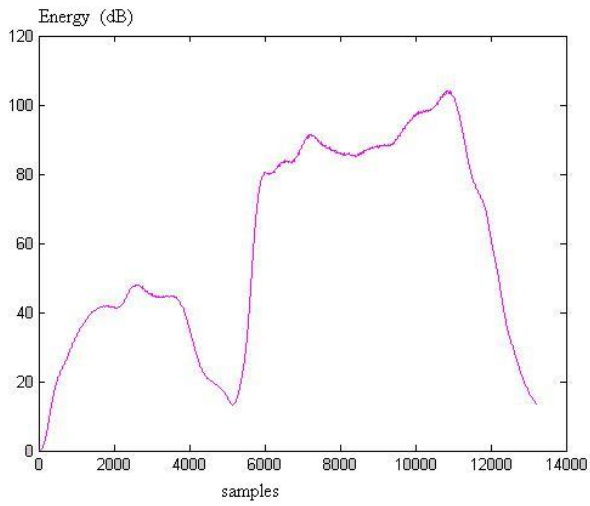

Fig. 5.

இங்கே

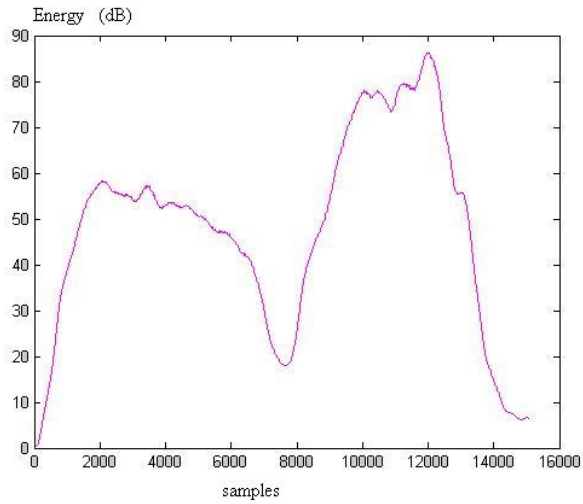

Fig. 6.

வாவா

Now consider the speech signal for the word **அம்மா**.

1) The speech wave is given as input to the software developed in Matlab. The energy values obtained are stored in e01.txt. Transfer energy values of 10th, 20th ,30th … points to a text file e02.txt.
2) The centered 21 point moving average method is used to smooth the energy curve of the speech signal. Store the resultant values in a text file e03.txt.
3) The turning points from minimum to maximum are found and the result is stored in the text file e04.txt.

The *x* and *y* values of the resulting pattern are given below:

10 : 0.026396
960 : 17.811204
3660 : 32.052359
3760 : 32.017806
3810 : 31.984550
4040 : 31.737289
5030 : 67.171043
5080 : 67.206326
6240 : 83.775497
6580 : 86.514993
6660 : 86.358709
7320 : 92.026408
12510 : 9.479558
12610 : 9.012534

4) Group the closely associated values sequentially and select the least value in each group removing the other values. The result is as follows

10 : 0.026396
960 : 17.811204
4040 : 31.737289
5030 : 67.171043
6240 : 83.775497
6660 : 86.358709
7320 : :92.026408
12610 : 9.012534

5) Check the relativity of the values from the second point onwards, and determine the turning points. Group the segments. Retain the first value of the each group and remove the rest. Store the resulting values. The result is

10 : 0.026396
960 : 17.811204
12610 : 9.012534

From the above, we now have two segments for the example considered. Thus, these two character sets in the word are considered.

In this case the output of the segmentation algorithm in the form of two segments is given as below.

Segment 1 : 10 to 960
Segment 2 : 960 to 12610

The starting and ending of the sample values of these segments are fed to the coding in Matlab. The Matlab coding converts these values into 18 cepstral coefficients for each segment. These cepstral coefficients are given as input to the BPN algorithm. The BPN algorithm gives the output for each segment represented by 18 cepstral coefficients into 12 cepstral coefficients corresponding to the format of the standard pattern. The outputs of the two segments obtained are given below.

Calculated cepstral coefficients for segment1: "அம் "

0.78805	0.04915	0.95098	0.73375	0.59913
0.75387	0.82703	0.75202	0.59981	0.73566
0.95105	0.05059			

Calculated cepstral coefficients for segment2: "மா"

0.71097	0.04915	0.93803	0.68716	0.56298
0.65186	0.95132	0.65274	0.56133	0.68641
0.94012	0.05075			

These output cepstral coefficients of each segment are compared with the cepstral coefficients of the standard patterns stored in the system.

The patterns whose cepstral coefficients are closer to the above values are picked up and given as the recognized characters "அம்" and "மா". The cepstral coefficients of these standard patterns are as given below.

Standard pattern for "அம்"

0.78717	0.05000	0.95000	0.73487	0.59868
0.75263	0.82812	0.75263	0.59868	0.73487
0.95000	0.05000			

Standard pattern for "மா"

0.71046	0.05000	0.93916	0.68779	0.56260
0.65131	0.95000	0.65131	0.56260	0.68779
0.93916	0.05000			

The same procedure is adopted for recognizing the characters of the other two words segmented from the speech input அம்மா இங்கே வாவா .

3. RESULTS AND ANALYSIS

The system developed was tested for all the 247 stand-alone characters of Tamil and the system worked very well with 100% accuracy. The system was also tested for segmenting characters from 149 words. Out of 149 words, 110 words were segmented correctly while 39 were not segmented correctly; percentage of success being 73.82.

In all the 149 words, there were 287 characters. When the system was tested for recognizing characters, 207 out of 287 characters were recognized correctly with a percentage success of 72.12.

The system was also tested for segmenting words from 9 sentences. The system worked well with 80.95% accuracy.

CONCLUSION

A procedure is proposed in this paper for recognizing spoken sentences and words in Tamil. A segmentation algorithm is proposed for the current situation and written in Matlab. The developed method is tested for 9 sentences and 149 spoken words. The segmentation algorithm for segmenting words works very well. The segmentation algorithm for segmenting spoken words also gives reasonably good results. In some cases it fails, because of the variations in the trained patterns and the test cases. This can be improved by enhancing the training of the patterns. The proposed scheme is tested for a given speaker and proved to be an effective approach for spoken Tamil speech recognition. However, this was tested for a given speaker. It is not tested for speaker independent. But it is possible to train the system for person independent application.

REFERENCES

1. M. Chandrasekar, M. Ponnavaikko. A Survey of methods used for Speech processing and the issues related to Indian Language Processing. Int. Conference on Spoken language processing. New Delhi, India, October 2002.
2. Davis K. H., Biddulph R., Balashek S. Automatic recognition of spoken digits. Journal of Acoust. Soc. of America, volume 24, pp. 637–642, 1952.
3. Olson H. F., Belar H. Phonetic Typewriter. Journal of Acoust. Soc. of America, 28(4), 1072–1081, 1956.
4. Fry D. B. Theoretical aspects of mechanical speech recognition. J. British Inst. Radio Engineer, 19(4), 211–218, 1959.
5. Forgie J. W., Forgie C. D. Results obtained from vowel recognition Computer program. J. Acoust. Soc. of America. 31(11), 1480–1489, 1959.
6. Sakai T., Doshita S. The phonetic typewriter. Information Processing. Proc. IFIP Congress, Munich, 445–450, 1962.
7. T. B. Martin, A. L. Nelson, H. J. Zadell. Speech recognition by feature extraction techniques. Tech. Report AL-TDR-64-176, Air Force Avionics Lab, 1964.
8. T. K.Vintsyuk. Speech discrimination by dynamic programming. Kibernetika, 4(2), 81–88, January-February, 1968.
9. D. R. Reddy. An approach to computer speech recognition by direct analysis of the speech wave. Tech. Report C549, Computer Science Dept., Stanford Univ., September 1966.
10. V. M. Velichko, N. G. Zagoruyko. Automatic recognition of 200 words. Int. J.Man-

Machine Studies, 2:223, June 1970.
11. H. Sakoe, S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoustics, Speech, Signal Proc., ASSP 26(1), 43–49, February 1978.
12. F. Itakura. Minimum prediction residual applied to speech recognition. IEEE Trans. Acoustics, Speech, Signal Proc., ASSP 23 (1), 67–72, February1975.
13. C. C. Tappert, N. R. Dixon, A. S. Rabinowitz, W. D. Chapma. Automatic recognition of continuous speech utilizing dynamic segmentations, dual classification, sequential decoding and error recovery. Tech. Report TR-71-146, Rome Air Dev. Cen, Rome, NY, 1971.
14. F. Jelinek, L R. Bahl, R. L Mercer. Design of a linguistic statistical decoder for the recognition of continuous speech. IEEE Trans. Information Theory, 1T-21, 250–256, 1975.
15. F. Jelinek. The development of an experimental discrete dictation recognizer. Proc. IEEE, 73(11), 1616–1624, 1985.
16. L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, J. G. Wilpon. Speaker independent recognition of isolated words using clustering techniques. IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-27: 336–349, August 1979.
17. H. Sakoe. Two level DP matching – a dynamic programming based pattern matching algorithm for connected word recognition. IEEE Trans. Acoustics, Speech, Signal Proc., ASSP 27, 588–595, December 1979.
18. J. S. Bridle, M. D. Brown. Connected word recognition using whole word templates. Proc. Int. Acoust. Autumn Conf., 25–28, November 1979.
19. C. S. Myers, L. R. Rabiner. A level building dynamic time warping algorithm for connected word recognition. IEEE Trans. Acoustics, Speech, Signal Proc., ASSP 29: 284–297, April 1981.
20. C. H. Lee, L. R. Rabiner. A frame synchronous network search algorithm for connected word recognition. IEEE Trans. Acoustics, Speech, Signal Proc., 37(11), 1649–1658, November 1989.
21. L. Rabiner, B. Juang, S. Levinson, M. Sondhi. Recent developments in the application of hidden Markov models to speaker-independent isolated word recognition. In Proc. of IEEE ICASSP-85, 9–12, Tampa, Florida, 1985.
22. J. Furguson, editor. Hidden Markov models for speech. IDA Princeton, NJ 1980.
23. L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. of IEEE, 77(2): 257–286, February 1989.
24. L. F. Baum, T. Petrie, G. Soules, N. Weiss. A maximization technique occurring statistical analysis of probablisitic functions of Markov chains. Ann. Math. Stat., 41, 164–171, 1970.
25. L. R. Liporace. Maximum likelihood estimation multivariate observations of Markov sources. IEEE Trans Info. Theory. IT-28, 729–34, September 1982.
26. B. Juang. Maximum likelihood estimation for mixture multivariate stochastic observations of Markov chains. AT&T Technical Journal, 64(6), 1235–1250, Part 1, July–August 1985.
27. B. H. Juang, S. E. Levinson, M. M. Sondhi. Maximum likelihood estimation for multivariate mixture observations of Markov chains. IEEE Trans Information Theory, IT-32(2), 307–309, March 1986.
28. J. K. Baker. Stochastic modeling for automatic speech understanding. In D. R. Reddy, editor, Speech Recognition. Invited Papers for the IEEE Symp.1975.
29. L. Bahl et al. Automatic recognition of continuously spoken sentences from a finite state grammar. In Proceedings ICASSP, Tulsa, OK, 1978.
30. B. T. Lowerre, R. Reddy. The HARPY speech understanding system. In W. Lea, editor, Trends in Speech Recognition, 340–360. Prentice Hall, Englewood Cliffs, NJ, 1980.

31. L. R. Bahl, F. Jelinek, R. L Mercer. A maximum likelihood approach to continuous speech recognition. IEEE Trans. PAMI, 5(2), 179–190, March 1983.
32. F. K. Soong, E. F. Huang. A tree-trellis fast search for finding N-best sentence hypotheses. In Proc. ICASSP 91, 705–708, Toronto, May 1991.
33. D. Paul. Algorithms for an optimal A* search and linearizing the search in the stack decoder. In IEEE ICASSP-91, 693–696, Toronto, Canada, May 1991.
34. B. H. Juang, R. Perdue, D. Thomson. Deployable automatic speech recognition systems: Advances and challenges. AT&T technical Journal, 74(2), 1995.
35. T. Kawahara, C-H Lee, B-H. Juang. Key-phrase detection and verification for flexible speech understanding. In Proc. ICSLP-96, Philadelphia, PA, October 1996.
36. M. Rahim, C-H Lee, B-H. Juang. A study on robust utterance verification for connected digits recognition. J. Acoustical Society of America, 1997.
37. T. Kawahara, C-H. Lee, B-R. Juang. Combining key-phrase detection and subword-based verification for flexible speech understanding. In Proc. of ICASSP-97, Munich, April 1997.
38. Z. Harris. Methods in Structural Linguistics. University of Chicago Press, 1951. Later updated and published as Structural Linguistics in 1960 and 1974.
39. F. Jelinek, L. R. Baul, R. L. Mercer. Design of a linguistic statistical decoder for the recognition of continuous speech. IEEE, Trans. Information Theory, IT-21, 250–256, 1975.
40. B. Juang. Speech recognition in adverse environments. Computer Speech & Language, 5, 275–294, 1991.
41. Y. Chen. Cepstral domain stress compensation for robust speech recognition. In Proc. of ICASSP-87, 717–720, Dallas, Texas, April 1987.
42. R. M. Stern, A. Acero, F. H. Liu, Y. Ohshima. Signal processing for robust speech recognition. In Automatic Speech and Speaker Recognition-Advanced Topics. Lee, Soong, Paliwal (eds.), 357–384, Kluwer, 1996.
43. M. G. Rahim, B. H. Juang. Signal bias removal by maximum likelihood estimation for robust telephone speech recognition. IEEE Trans. SAP, 4(1), 19–30, January 1996.
44. M. J. F. Gales, S. J. Young. Robust speech recognition in additive and convolutional noise using parallel model combination. Computer Speech and language, 9, 289–307, 1995.
45. C. H. Lee, J. L. Gallvain. Baysian adaptive learning and MAP estimation of HMM. In C.H. Lee, F. K. Soong, K. K. Paliwal, editors, Automatic Speech and Speaker recognition: Advanced Topics, chapter 4. Kluwer Academic Publishers, 1996.
46. B. H. Juang, C. H. Lee, C. H. Lin. A study of speaker adaptation of the parameters of continuous density hidden Markov models. IEEE trans. Acoustic, Speech, Signal Proc., 39(4), 806–814, April, 1991.
47. A. Sankar, C. H. Lee. A maximum-likelihood approach to stochastic matching for robust speech recognition. IEEE Trans on Audio and Speech Forcasting, 4(3), 190–202, 1996.
48. B. Juang, W. Chou, C. H. Lee. Minimum classification error rate methods for speech recognition. IEEE trans. Speech and Audio Proc. T-SAP, 5(3), 257–265, May 1997.
49. B-H. Juang, S. Katagiri. Discriminative learning for minimum error training. IEEE Trans. Signal Proc. 40(12), 3043–3054, December, 1992.
50. Ji Ming, Peter O'Boyle, Marie Owens, F. Jack Smith. A Bayesian approach for building triphone models for continuous speech recognition. IEEE Trans. on Speech and Audio Processing, vol. 7, 678–684, 1999.
51. L. Lamel, J. L. Gauvain. Large-vocabulary continuous speech recognition: Advances and applications. Proceedings of IEEE, vol. 88, 1181–1200, 2000.

52. R. Lippmann. An introduction to computing with neural nets. IEEE ASSP Magazine, vol. 4, 4–22, April 1987.
53. R. Lippmann, W. Huang, B. Gold. A neural net approach to speech recognition. Int. Conf. ASSP, 99–102, 1988.
54. D. J. Burr. Experiments on neural net recognition of spoken and written text. IEEE Trans. ASSP, vol. 36, no. 7, 1162–1168, 1988.
55. Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., Lang, K. J. Phoneme recognition using time-delay neural networks. IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. 37(3), 328–339, 1989.
56. Levin, E. Word recognition using hidden control neural architecture. IEEE Conf. on Acoustics, Speech, and Signal Processing, ICASSP-90, vol. 1, 433–436, 1990.
57. Islam, R., Hiroshige, M., Miyanaga, Y., Tochinai, K. Phoneme recognition using modified TDNN and a self-organizing clustering network. IEEE Int. Symp. on Circuits and Systems, ISCAS 95, vol. 3, 1816–1819, 1995.
58. B. A. Pearlmutter. Learning State Space Trajectories in Recurrent Neural Networks. Neural Computation, vol. 1, 365–372, 1989.
59. Hasegawa H., Inasumi, M. Speech Recognition By Dynamic Recurrent Neural Networks. International Joint Conference on Neural Networks, IJCNN'93, vol. 3, 2219 –2222, 1993.
60. Jaing Minghu. Fast Learning Algorithms for Time-Delay Neural Networks Phoneme Recognition. Proc. ICSP, 730–733, 1998.
61. H. Iwamida, S. Katagiri, E. McDermott, Y. Tohkura. A hybrid speech recognition system using HMM with an LVQ trained code book. J. Acoust. Soc. Japan, vol. 11, no. 5, 277–286, 1990.
62. Katagiri.S, Lee. C. H. A new hybrid algorithm for speech recognition based on HMM segmentation and learning vector quantization. IEEE Trans. on Speeech and Audio Processing, vol. 1, no. 4, 421–430, 1993.
63. C. Dugast, L. Devillers, X. Aubert. Combining TDNN and HMM in a hybrid system for improved continuous speech-recognition system. IEEE Trans. on Speeech and Audio Processing, vol. 3, no. 1, 1994.
64. G. Zavaliagkos, Y. Zhao, R. Schwart, J. Makhoul. A hybrid segmental neural net/hidden Markov model system for continuous speech recognition. IEEE Trans. on Speeech and Audio Processing, vol. 2, no. 1, 151–160, 1994.
65. J. Suzuki, K. Nakata. Recognition of Japanese vowels-preliminary to the recognition of speech. J. Radio Res. Lab, 37(8), 193–212, I961.
66. K. Nagata, Y. Kato, S. Chiba. Spoken digit recognizer for Japanese language. NEC Res. Develop., 6, 1963.
67. Jean-Marc Boite, Christophe Ris. Development of a French Speech Recognizer Using a Hybrid HMM/MLP System. ESANN'1999 proceedings - European Symposium on Artificial Neural Networks, Bruges (Belgium), 441–446, 21-23 April 1999.
68. Solomon Teferra Abate, Wolfgang Menzel. Syllable-Based Speech Recognition for Amharic. Proc. Of the 5th workshop on important unresolved matters, 33–40, Prague, Czeech Republic, June, 2007.
69. C. Chandra Sekhar, J. Y. Siva Rama Krishna Rao, Recognition of Consonant-Vowel (CV) units of speech in Indian languages, Proc. National seminar on Information Revolution and Indian Languages, Hyderabad, 22.1–22.6, Nov. 12-14, 1999.
70. Suryakanth V. Gangashetty, B. Yegnanarayana, Neural network models for recognition of Consonant-Vowel (C^nV) utterances, INNS-IEEE International Joint Conference on Neural Networks, Washington DC, July 14-19, 2001.

71. A. Nayeemulla Khan, Suryakanth V. Gangashetty, S. Rajendran. Speech database for Indian languages - A preliminary study. Proc. Int. Conf. Natural Language Processing, IIT Bombay, Mumbai, 295–301, Dec. 2002.
72. S. R. M. Prasanna, J. M. Zachariah, B. Yegnanarayana. Begin-end detection using vowel onset points. In Proc. Workshop on Spoken Language Processing, TIFR, India, Jan. 2003.
73. S. V. Gangashetty, K. Sreenivasa Rao, A. Nayeemulla Khan, C.Chandra Sekhar, B.Yegnanarayana. Combining evidence from multiple modular networks for recognition of consonant-vowel units of speech. Int. Joint Conf. Neural Networks, Portland, USA, July 2003.
74. Samudravijaya K, P. V. S.Rao, S. S. Agrawal. Hindi Speech Database. Proc. Int. Conf. on Spoken Language processing, Beijing, China, October 2000.
75. Samudravijaya K. Computer Recognition of Spoken Hindi. Proc. Int. Conf. Speech, Music and Allied Signal Processing, Thiruvananthapuram, 8–13, Dec. 2000.
76. Samudravijaya K. Hindi Speech Recognition. J. Acoustic Society of India, vol. 29, issue 1, 385–393, 2001.
77. Samudravijaya K. Durational Characteristics of Hindi Phonemes in Continuous Speech. Sept. 2003.
78. Chalapathi Neti, Nitendra Rajput, Ashish Verma. A large vocabulary continuous speech recognition system for Hindi. IIT, Bombay, Jan. 2000.
79. Paul Mermelstein. Automatic segmentation of speech into syllabic units. J. Acoust. Soc. Am., vol. 58, no. 4, October 1975.
80. Andre G. Adamiand and Hynek Hermansky. Segmentation of speech for speaker and language recognition. EUROSPEECH, Geneeva, 2003.
81. Ramil G. Sagum, Ryan A.Ensono, Emerson M. Tan, Rowena Christiana L. Guevara. Phoneme alignment of Filipino speech corpus. Conference on convergent technologies for Asia Pacific Region, vol. 3, 964–968, Oct. 2003.
82. A.Lipeika, G.Tamulevicius. Segmentation of words into phones. ISSN 1392-1215, Electronics and Electrical Engineering, 1(65), 2006.
83. R. W. Schafer, L. R. Rabiner. Digital representations of speech signals. Proc. IEEE, vol. 63, no. 4, April 1975.