Asian Journal of Information Technology 5 (1) : 38-43, 2006

Integrating Language Independent Segmentation and Language Dependent Phoneme Based Modeling for Tamil Speech Recognition System

[1]Saraswathi, S., [2]T.V. Geetha and [3]K. Saravanan
[1,2,3]Department of Computer Science and Engg., Anna University, Chennai, India

Abstract: This study describes the work done in performing speech recognition for Tamil language, using a new approach of text independent speech segmentation, with a phoneme based language modeling. A performance improvement in the recognition is possible by combining the results of segmented phoneme sequence with phoneme based language model. The results are promising since the method gives a 75.8% of correct segmentation and better word recognition rate for Tamil language.

Key words: Language modeling, perceptual linear predictive coding, phoneme segmentation, speech recognition

INTRODUCTION

HERE are various approaches to implement Tamil speech recognition system. One approach suggested by Yegnanarayana *et al.*,[1] was based on hidden Markov Model and was for tasks considering a small vocabulary (10-100 words). A multilingual speech recognition system for Hindi, Tamil and Indian accented English was proposed by using classification and Regression trees to predict phones[2]. Accurate segmentation of speech signals is an important component of a speech recognition system. Common approaches to speech signal segmentation are based on application of viterbi and forward-backward algorithm to a corpus of transcribed speech signals[3]. A novel method was proposed by Murthy, Hema A. *et al.,*[4] for segmentation of continuous speech signals using modified group delay and MFCC feature based on HMM for Tamil and Telugu languages. A new approach is proposed in this study to implement speech recognition system for Tamil language, by using an algorithm, which will perform phonetic segmentation of speech without prior knowledge of the phoneme sequence contained in the waveform. This approach does not rely on an externally supplied transcription of the sentence for determining phoneme boundaries as proposed in[5]. The approach described in this study could be used for tasks such as text independent segmentation of multi-lingual speech, which is a fundamental element of phoneme-based speech recognition system. The phoneme based language modeling for Tamil language is used in the segmentation process to avoid certain ambiguities that arise in the recognition of phoneme sequences.

Text independent speech segmentation task proposed by[6] is carried out in two fundamental steps: Speech preprocessing followed by phoneme boundary detection. This approach led to some under segmentation of data in Tamil language. So a recursive method of re-segmenting the under segmented signal sequence is proposed in this work. This two-level segmentation is language dependent and gave good segmentation results for Tamil language.

The phoneme classification inaccuracy at the segmentation of phonemes is a major weakness in the speech recognition systems[7]. Usually three types of phoneme classification errors appear: (1) phoneme addition, (2) phoneme deletion (3) phoneme replacements. A better performance is expected if a language model is adopted in a recognition system for post-processing phoneme estimates and making corrections with a set of explicit rules of the language being used. The recognition of natural speech requires making optimum use of all available knowledge sources.

The contribution of linguistics to speech recognition will mainly materialize in terms of the knowledge that can be applied to reduce errors after the recognition process is completed. In many cases applying simple grammatical constraints will essentially improve the performance of a speech recognizer.

Achieving consistent performance in varied application environments will also play an important role, among other things, in the user acceptance of the speech recognition system[8].

In this study the results of applying a rule-based phoneme language model to correct the three types of phoneme classification errors recognized by the segmentation algorithm was found to improve the performance of Tamil speech recognition system. Section 2 describes the steps involved in the segmentation algorithm. Section 3 describes the use of language modeling approach to improve the efficiency of the recognition system. The last section includes the results and improvements observed in the existing system.

SPEECH SEGMENTATION

The text independent speech segmentation algorithm[6] proposed by Guido Aversano gives good segmentation results (73.5%). When the algorithm was applied on Tamil speech corpus, there were some under segmented signal sequences i.e. syllables, in the segmented output sequence. So the one level segmentation algorithm[6] proposed by Guido Aversano was modified to a two-level segmentation algorithm, in order to segment only the under segmented syllable sequences in Tamil language.

The segmentation process is done by first preprocessing the speech signal, followed by the detection of phoneme boundaries. The phoneme sequences, which are under segmented, are then again re-segmented by detecting the peak points in those sequence whose values are within a particular threshold. The steps involved in the two-level segmentation process are discussed in this section

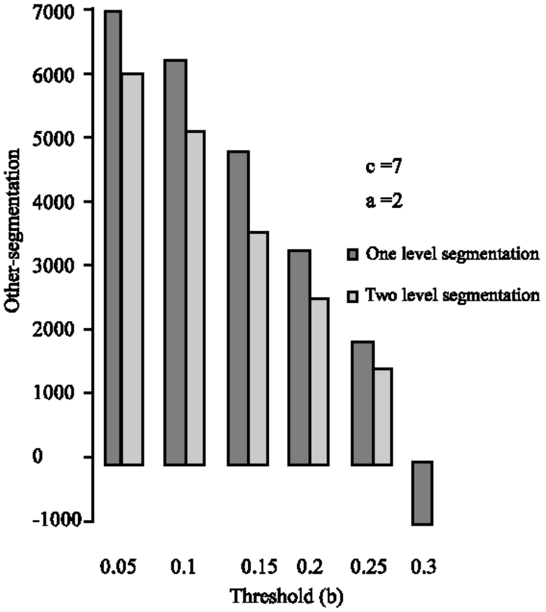

Fig. 1: Over-segmentation as function of the threshold b, for a fixed a=2 and c=7

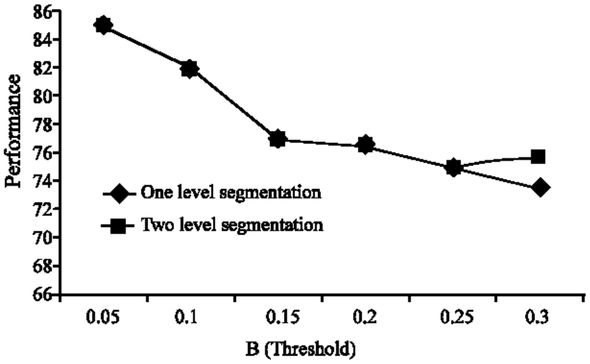

Fig. 2: Detection of performed as function of the threshold b, for a fixed a=2 and c=7

Speech preprocessing : A physical process in which movements of the articulatory apparatus produce non-stationary sound waves creates the speech signal. The signal sampled at 16kHz, is decomposed into a sequence of overlapping frames. Each frame corresponds to 15 ms (240 samples) of signal with a frame overlap 7.5 ms (120 samples). The samples are weighted by a Hamming window to avoid spectral distortions[9]. Each frame is then passed through a perceptual –based analysis, which includes critical-band resolution, equal loudness pre-emphasis and intensity –loudness compression of the Fourier spectrum. This crude perceptually modified spectrum can be used for the detection of phoneme boundaries without application of any temporal filtering and predictive modeling process – as for example, in the PLP and RASTA-PLP preprocessing schemes[10] also based on perceptual analysis. This procedure, which uses the concepts from psychophysics of hearing in order to obtain an estimate of the auditory spectrum, gives a relatively small number of features for each frame. Each feature quantifies the spectral energy found in a particular frequency interval. For the experiments reported in the present study , using an algebraic sum over the groups of three parameters, the number of features was further reduced to five. The output of the preprocessing phase is, therefore, a collection of time sequences, where the output of the analysis performed on a single frame of the speech signal is represented at each point. This collection is denoted by $\{x_i[n]\}^{k}_{i=1,} n = 1 \ldots N$, where N is the total number of frames, and k is the number of time-sequences. The value of k was set to five to obtain better results.

Phoneme boundary detection: The one-level segmentation algorithm[6] is regulated by three operational parameters *a*, *b* and *c*. The first two are related to the determination of the jumps for each sequence of parameters. The parameter *a* is a measure of the number of consecutive points in the sequence used to estimate the height of a jump, when this height exceeds a certain relative threshold (*b*). The parameter *c* is used in the final step, for adjusting the width of the subintervals in which the algorithm searches for the middle point of quasi-simultaneous jumps.

The algorithm was tested on subset of Tamil speech corpus collected from doordarshan News corpus. A collection of 200 sentences was used, representative of 20 speakers (10 male and 10 female). The best results were obtained for *a*=2, *b*=0.3 and *c*=7 as shown if Fig. 1 and Fig. 2. But for other values of *b* and *c* below this range over-segmentation of speech signal occurred , while values above this led to under segmentation. Setting *b*=0.1 led to best performance (82%) of the segmentation algorithm, but however also led to over segmentation. Performance measure of 73.5% was obtained by applying the one-level segmentation algorithm on Tamil speech corpus with values of *a*=2, *b*=0.3 and *c*=7. One of the main shortcomings in applying this algorithm for Tamil language was that, it leads to under segmentation of some signal sequences, resulting in syllables instead of phonemes. In order to overcome this problem the two-level segmentation algorithm was designed.

As discussed in[6] the first step in phoneme boundary detection is to determine the peak points, jumps, in the time sequences, where the acoustic features obtained from the pre-processing phase changes significantly and quickly. This is calculated using the jump function defined as follows.

$$J_i^a[n] = \left| \sum_{m=n-a}^{n-1} \frac{x_i[m]}{a} - \sum_{m=n+1}^{n+a} \frac{x_i[m]}{a} \right| \qquad (1)$$

a represents the number of frames, before and after n, where the jump function is computed. A valid jump event in the i-th sequence is defined as a peak of the function $J_i^{(a)}[n]$ whose height exceeds a certain threshold *b*.

The segmentation algorithm[6] takes care of combining a unique indication of phoneme boundary, the jump events which are detected around the same frame 'n' in 'k' distinct time sequences. The jumps do not occur simultaneously within each of the k-time sequence. So a fitting procedure described in this work was introduced to place the sequence boundary in the middle of a cluster of quasi-simultaneous jumps. The series of detected peaks are store in acc[n] where n ranges from 1 to N and N is the number of peaks detected in the signal.

In our modification to the original segmentation algorithm instead of considering all peaks, a phoneme boundary is placed in each frame only if the peak is greater than a threshold value (thresh1). This threshold has been defined as follows.

$$\text{Thresh 1} = \frac{\max(\text{acc}[n]) + \min(\text{acc}[n])}{2} \qquad (2)$$

The peak is detected at the point n where acc[n] is greater than the threshold value (thresh1).

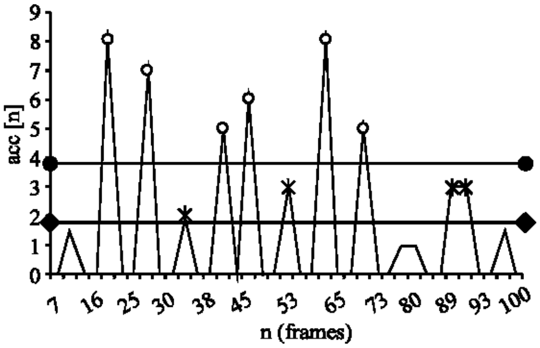

Fig. 3: A typical acc[n] sequence - segmentation point after 1-level segmentation indicated by O points included after 2- level segmentation indicated by *

The other major modification to the algorithm is the introduction of a post segmentation step in which a second level of segmentation is carried out. This second level of segmentation is applied to only those segmented signal sequences that do not match corresponding Tamil phonemes. This is detected by comparing PLP values of the segmented signals with PLP values of stored Tamil phonemes. In addition any segmented signal having duration greater than maximum allowed duration of Tamil phoneme (20 ms) were assumed to be syllable patterns. These segments were also candidates for further segmentation.

A new threshold (thresh2) was assumed which was fixed at half of thresh1, the threshold used for first level of segmentation. In the second level of segmentation peaks of the segments whose range was between thresh1 and thresh2 were detected and the syllable segments were segmented.

Figure 3 shows the peaks detected using thresh1 and thresh2. The algorithm was tested for different values of *a*, *b* and *c*. The parameters *a*=2, *b*=0.3 and *c*=7 gave better results for the two-level segmentation algorithm, with no under segmentation of signal sequences. The two-level segmentation algorithm was able to correctly detect 75.8% of the phoneme transitions. The performance of the segmentation algorithm with two-level segmentation showed an improvement of 2.3% over the one-level segmentation algorithm as shown in Fig.2.

The output of the segmentation algorithm is a sequence of phonemes. The phonemes were mapped onto their corresponding Tamil graphemes based on their PLP characteristics. There were many errors detected in the grapheme output due to insertion, deletion and replacement of consonants and vowels in the words. As shown in Table 1 most of the errors were due to insertion of many vowels in the place of one, i.e, doubling of vowels occur which is not possible in Tamil. This was mainly due to overlapping of frame sequences. Moreover there were also large number consonant replacement errors in grapheme sequence. This was due to similar characteristics of certain sequence of phonemes in Tamil like /–/, /‹/ and /n/.

In order to reduce such errors a rule based phoneme language model was designed for Tamil language as discussed in the next section.

RULE BASED PHONEME LANGUAGE MODEL FOR TAMIL LANGUAGE

At the lexical level, phoneme classification errors may violate phonological rules and vowel harmony rules in Tamil language. Consequently the application of these rules to recognized word candidates in a post-processing manner would correct some phoneme classification errors[11]. The phonological rules for Tamil language was obtained by applying phone based language models on the Tamil phoneme sequence.

A Tamil text with 50,000 words was collected from News study s. A grapheme to phoneme converter was used to convert the text data to sequence of phonemes. In order to implement a Tamil Grapheme to Phoneme system many language specific problems need to be solved. Various approaches like dictionary-based and rule-based grapheme to phoneme converters exist. The dictionary-based approach[12] requires a large dictionary and does not deal with unregistered word. The rule-based approach makes use of the features of Tamil language to detect the corresponding phonemes for the graphemes based on their position, type and acoustic characteristics[13]. The rule-based approach is used to perform grapheme to phoneme conversion for Tamil language.

The rule-based approach is in general based on the characteristic of Tamil language[14]. In Tamil there are in general twenty-nine phonemes that have a matching grapheme representation. However Tamil allows multiple allophonic variations for a single grapheme. Considering all allophonic variations there are fifty phonemes for Tamil language. In Tamil the phonemes are divided into three groups namely 1. Vowels, 2. aytam and 3. Consonants. There are ten vowels five short and five long. There are eighteen consonants in Tamil. Based on the rules proposed by Pon. Kothandaraman[15], the rule based grapheme to phoneme converter was designed for Tamil language.

The rule-based grapheme to phoneme converter is used to convert the news study text corpus with 50,000 words to their corresponding phoneme sequences. The phoneme sequence contained about 2 lakh phonemes. Phoneme based language modelling is used to get information regarding the possible phoneme occurrences in the start, middle and end of word in a particular language[16]. Bigram and Trigram based language models were applied over the phoneme sequences to perform an analysis of the possible phoneme sequence occurrence in Tamil speech corpus. Based on the characteristics of Tamil language and the study of the bigram and trigram based phoneme language models phonological rules were developed. The phonological rules based on characteristics of Tamil language is listed below

- Phonological constraints in three positions of the word, namely, at the beginning, in the middle and at the end of the word. For instance in Tamil no two consonants can appear together in the beginning of the word.
- Based on vowel harmony rule, the vowels /…/, /‡/, /†/ and /„/ do not occur in the middle of a simple word and no two vowels occur together in Tamil language.
- Two long vowels ai and au exist in Tamil. Modern scholars call them diphthongs. The vowels /†/, /„/ and /au/ do not occur in the non initial position of a simple word.
- The sequence CV is considered as one letter, called as uyirmey in Tamil. The first character in a word can be either a Vowel or a uyirmey in Tamil language.
- Due to contact with Sanskrit language, certain phonemes /j/, /™/, /k™/ of Sanskrit language also exist along with Tamil phonemes.
- The phoneme aytam is used in modern Tamil, described as a glottal voiceless fricative. The function of the phoneme is to fricativize the stops that follow it. It occurs between a short vowel and a stop consonant
- The consonant /p/ a labial voiceless stop has tree allophones [p] when exiting in initial position and in gemination [b] when preceded by a nasal [Ø] in intervocal position between consonants and vowels.

The phonological rules based on the application of language models to Tamil phoneme sequences is listed below

- Vowel Constraints occur at the end of words. The phoneme /i/ does not occur at the word endings.
- Consonants had the restrictions of having only the phonemes /m/,/l/, /!/, /‹/ and /r/ to occur most of the time at the end of the words. The percentage of occurrence being 31% for /m/, 25% for /l/, 18% for /!/, 16% for /‹/ and 10% for /r/.
- The phonemes /Œ/, / L/, /—/ and /_t/ occurs only in the middle of the words.
- All consonants do not occur in gemination. Certain phonemes like /_l/ and /r/ do not geminate.
- The consonant /t/ and /p/ only occurs in the beginning and middle of the word.

By applying the above phonological rules on the segmented phoneme sequence most of the errors were corrected. Many errors due to doubling of vowels and consonants were detected and modified. Errors due to consonant replacement were detected. Most of the errors due to the deletion of vowels and consonants were not detected using this approach.

RESULTS

Tamil texts of about 200 sentences were read aloud and the two-level segmentation algorithm was used to segment them. The test material contained a total of 6200 phonemes and the recogniser had technical prerequisites to recognize of about 4350 of them, which gives the percentage of correctly recognized phonemes of 70.16%. The distribution of the recognition errors is 16.1% for deletion, 43.5% for addition and 40.4% for replacement.

As shown in Table 1, the distribution of phoneme classification errors in the test words differs in different positions of the word. In all there are 211 phoneme classification errors of which 59 appears at the beginning of the word, 130 in the middle of the word and 22 at the end of the word, respectively. Fig. 4 shows the improvement in the performance of the recognition rate by applying the phoneme based language models to the segmented data. The total percentage of the phoneme classification errors corrected by applicable rules of the language model to the segmented phoneme sequence is 60.2%.

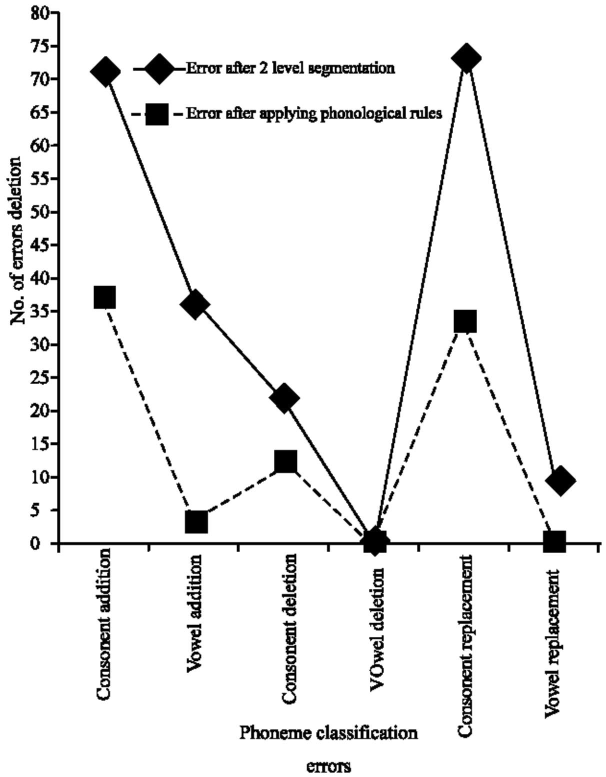

Fig.4: Comparison of the error detected from the segmentation algorithm and after applying the phonological rules on the segmented phoneme sequence

Table 1: Distribution of phoneme classification errors in three word positions and the number of phoneme classification errors corrected by applicable rule of a language model developed

	Word position				
Error type	B	M	E	N	Error corrected by rules
Consonant addition	18	48	05	71	34
Vowel addition	12	24	-	36	34
Consonant deletion	2	9	11	22	10
Vowel deletion	-	-	-	-	-
Consonant replacement	25	44	4	73	40
Vowel replacement	2	5	2	9	9
Total	59	130	22	211	127

Symbols: B=at the beginning of the word, M=in the middle of the word, E=at the end of the word, N= Total no. of errors

CONCLUSIONS

In this study a speech recognition system for Tamil language is implemented using a new method of two-level speech segmentation algorithm and applying the phoneme based language modeling on the output of the segmentation algorithm to reduce the recognition errors. The speech segmentation method is based both on a new pre-processing approach and a new two level segmentation algorithm. The pre-processing, based on concepts from psychoacoustics, is able to extract robust acoustic features, which capture information about phoneme transitions. The two-level segmentation algorithm correctly detected about 75.8% of the phoneme transition when compared to the one-level segmentation algorithm, which detected only 73.5% of the phoneme transition. The phonological rules for Tamil language obtained by applying phone based language models on the Tamil phoneme sequence, was then applied to the segmented sequences to obtain a recognition rate of 70.16% for Tamil language.

REFERENCES

1. Yegnanarayana, B and A. Nayeemullah Khan., 2001. Development of a speech recognition system for Tamil for restricted small tasks, In Proc. NCC , India.
2. Udhyakumar, N., R. Swaminathan. and S.K. Ramakrishnan., 2001. Multilingual Speech Recognition for information retreival in Indian context , Proceedings of the Student Research Workshop at HLT-NAACL.

3. Demuynck, K. and L. Tom, 2002. A Comparison of Different Approaches to Automatic Speech Segmentation. In Proc. 5th International Conference on Text, Speech and Dialogue, Brno, Czech Republic. pp: 277-284,
4. Hegde, R.M., M.A. Hema. and G.R. Venkata, 2004. Continuous speech recognition using joint features derived from the modified group delay function and MFCC, In INTERSPEECH- pp: 905-908.
5. Pellom, B.L. and J.H.L. Hanse, 1998. Automatic segmentation of speech recorded in unknown noisy channel characteristic, Speech Communication, Phil. Trans. Roy. Soc. London, pp: 529-551.
6. Aversano, G., E. Anna, E. Antonietta and M. Maria, 2001. A new text independent method for phoneme segmentation, In Proceedings of the IEEE International Workshop on Circuits and Systems, 2: 516-519.
7. Ray, A and A.K. Datta, 1994. Hierarchical perception linked model for machine recognition of vowels in telugu, Acustica 80: 406-412
8. Viglione, S.S., 1985. Speech recognition-projection and prognosis,Official Proceedings of Speech Tech '85. Voice Input/Output Application show and conference, Media dimensions, New York, NY,USA,pp: 17-19.
9. Rabiner L and R.W. Schafer, 1978. Digital Processing of Speech Signals, Prentice hall.
10. Hermansky H., 1990. Perceptual Linear Predictive (PLP) Analysis of Speech, J. Acoustic Society, America, 87: 1738-1752
11. Vayrynen, P., P. Johannes. and S. Tapio, 2000. Enhancing Phoneme recognizer performance with a simple rule based language model, Proc. Finnish Artificial Intelligence Days, Espoo, Finland, pp: 171 - 178.
12. Mittapiyanurak, P., C. Hansakunbuntheung, V. Tepprasit. and V. Sornlertlamvanich, 2000. Issues in Thai text to speech synthesis: The NECTEC Approach, Proceedings of NECTEC Annual conference, Bangkok, Tailand, 2000, pp:483-495
13. Choudhury, M., 2003. Rule based grapheme to phoneme mapping for Hindi speech synthesis, 90th Indian Science Congress of ISCA, Bangalore.
14. Arden, A.H., A.C. Rev ang A. Clayton, 1969. A progressive grammar of the Tamil Language , Madras
15. Kothandaraman, P., 1997. A grammar of contemporary literary tamil ,international institute of Tamil Studies, Chennai.
16. Magimai.-Doss, M., A. Todd, H.B. Stephenson, and B. Samy, 2003. Phoneme-grapheme based speech recognition system 2003 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU '03), St. Thomas, U.S. Virgin Islands, pp: 94-98.