A SYLLABLE BASED CONTINUOUS SPEECH RECOGNIZER for TAMIL

Lakshmi A, Hema A Murthy

Department of Computer Science and Engineering
Indian Institute of Technology, Madras
{*lakshmi*,*hema*}@lantana.tenet.res.in

Abstract

This paper presents a novel technique for building a syllable based continuous speech recognizer when unannotated transcribed train data is available. We present two different segmentation algorithms to segment the speech and the corresponding text into comparable syllable like units. A group delay based two level segmentation algorithm is proposed to extract accurate syllable units from the speech data. A rule based text segmentation algorithm is used to automatically annotate the text corresponding to the speech into syllable units. Isolated style syllable models are built using multiple frame size (MFS) and multiple frame rate (MFR) for all unique syllables by collecting examples from annotated speech. Experiments performed on Tamil language show that the recognition performance is comparable to recognizers built using manually segmented train data. These experiments suggest that system development cost can be reduced by using minimum manual effort if sentence level transcription of the speech data is available.

Index Terms: syllable based speech recognition, acoustic group delay segmentation, text segmentation, annotation.

1. Introduction

The current day challenge in building a continuous speech recognition system is the cost of obtaining the necessary annotated transcribed train data, which is an expensive process in terms of both manpower and time. A lot of research has gone into unsupervised training techniques using untranscribed data [1]. These ideas greatly rely on a trusted recognizer, huge amounts of acoustic data and a good language model. Over the last decade the importance of syllable based speech recognition systems have been realized as syllables are known to be acoustically and perceptually stable units[2]. Unsupervised incremental clustering techniques for training syllable based recognizers are tried out in [3]. These techniques allow syllable clusters to be automatically formed using an incrementally built recognizer and then the clusters are manually labeled. The drawback of this approach is that different similar sounding syllables often get clustered together resulting in a difficulty in labeling the clusters themselves. Less accurately clustered data and the manual labour of labeling the clusters make these techniques less preferable.

In this paper, we assume that sentence level transcriptions are available for the speech signal. A syllable level segmentation algorithm is applied to the acoustic signal. The sentence text transcriptions are also syllabified using a text segmentation algorithm. These units are matched to generate the automatic annotations.

Earlier work in [4] shows that, even though the group delay based segmentation algorithm identifies segment boundaries quite accurately, 30% of the syllable boundaries are missed. [5] modifies the algorithm which results in an improvement of 10% boundary detection. The limitation of the boundary detection algorithms is due to the fact that the accuracy of the boundaries depend on the characteristics of the speech signal, especially the variable syllable rate across an utterance. We propose a two level segmentation algorithm which utilizes the syllable duration information to resegment the units if necessary.

The text segmentation is based on the linguistic rules derived from the language. Any syllable based language can be syllabified using these generic rules. To make the text segments exactly equivalent to the speech units, few language specific rules will be useful. After the text is segmented, the segmented waveform and segmented text are matched. Multiple realisations of each unique syllable are extracted and HMMs are trained for each of the syllables. Acoustic models are built using the multiple frame sizes (MFS) and multiple frame rates (MFR) based feature extraction technique[6]. This accounts for the spectral variations along time and also the sparsity in train data. The performance of this automatically annotated recognizer (AAR) is compared with that of a conventional HMM based continuous speech recognizer (CCSR). While the AAR recognizer uses syllabified text and continuous speech with segment boundary information to build the syllable models for the continuous speech recognizer, the CCSR transcribes continuous speech by automatically detecting and aligning boundaries.

The paper is organised as follows. Section 2 gives the details of the speech segmentation algorithm including the baseline group delay based segmentation and the modified two level segmentation. Section 3 gives the details of the linguistic rules applied to generate segmented text. The experimental setup including the acoustic model generation, testing and results are discussed in section 4. This section also explains the conventional HMM based continuous speech recognizer. Section 5 discusses the important observations.

2. Group delay based segmentation of syllable units

In this section, we review the baseline algorithm for syllable segmentation [5] and then discuss the proposed modifications.

2.1. Baseline segmentation algorithm

The segmentation algorithm uses a minimum phase signal derived from the short term energy (STE) function as if it were a magnitude spectrum. The high energy regions in the STE function correspond to the syllable nuclei while the valleys at both ends of the nuclei determine the syllable boundaries. The algorithm is as follows.

1. Let x(n) be a digitized speech signal of a continuous speech

utterance.

2. Compute the STE function E(m), using overlapped windows. Since this is viewed as an arbitrary magnitude spectrum, let it be denoted as $E(K)$.
3. Construct the symmetric part of the sequence by producing a lateral inversion of this sequence about the Y-axis. Let it be denoted as $\tilde{E}(K)$.
4. Compute $(1/\tilde{E}(K))$. Let the resultant sequence be $\tilde{E}^i(K)$.
5. Compute IDFT of the sequence $\tilde{E}^i(K)$. This resultant sequence $\tilde{e}(n')$ is the root cepstrum and the causal portion of it has the properties of a minimum phase signal.
6. Compute the minimum phase group delay function of the windowed causal sequence of $\tilde{e}(n')$. Let this sequence be $\tilde{E}_{gd}(K)$. Let the size of the window applied on this causal sequence, that is, the size of the cepstral lifter be N_c.
7. The location of the positive peaks in the minimum phase group delay function $\tilde{E}_{gd}(K)$ approximately correspond to syllable boundaries.

A factor called cepstral lifter (N_c) determines the frequency resolution in the group delay spectrum. As the duration of an utterance can vary considerably across different utterances, and, the resolution of segmentation depends crucially on the lifter window, the lifter window is made to be a function of STE. It is defined as

$$N_c = \frac{Length\ of\ the\ energy\ function}{Window\ scale\ factor(WSF)} \quad (1)$$

where WSF is an integer > 1. The length of the STE function corresponds to the number of samples in the STE function and WSF represents the scale factor used to truncate the cepstrum. If N_c is high, resolution is also high, i.e., it can resolve closely spaced syllables. Choice of WSF and ultimately N_c depends on the syllable rate.

In [7] it was observed that the segmentation algorithm performed poorly during the segmentation of semivowels, fricatives and long silences. A significant improvement in performance is observed when the speech signal is first band pass filtered and then subjected to group delay segmentation. However, spurious boundaries are observed at the beginning of a nasal consonant if the resolution is high. Long silences also result in spurious boundaries. A general long silence removal algorithm needs threshold manipulation and is likely to remove low energy units like fricatives. The low pass filter for removing high frequency fricatives removes the fricatives as such instead of correcting the boundaries. The undetected boundaries in words with semivowels due to their high energy is resolved using the band pass filter. These measures do not guarantee the identification of all the boundaries correctly. All issues revolve around the fact that the whole sentence is segmented using a single resolution. If the frequency resolutions can be varied within a sentence, better boundaries can be detected. Long silences form the first level of boundaries with a low resolution scale factor. Each individual segments can then be resegmented using high resolution scale factors to get accurate boundaries. This idea is central to the two level segmentation algorithm described below. The fricative removal problem is also rectified as no filtering is needed. On acoustic data of shorter length, the fricative boundaries are better identified. The semivowel boundaries are more accurately extracted using high resolution scale factors on the smaller speech units obtained after the first level of segmentation.

Table 1: Recognition with single and two-level segmentations

WSF Value	4	5/4	6/3
Recognition Performance	37.81%	41.48%	40.43%

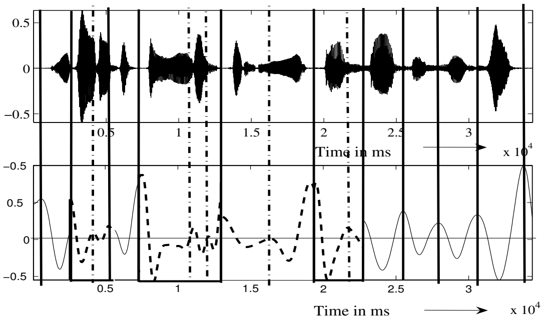


Figure 1: Two level segmentation of a sample sentence using WSF 5 (shown with solid vertical lines) followed by WSF 4 (shown with dotted vertical lines). The group delay function for the polysyllabic units with WSF 4 is placed on top of the group delay function for WSF 5.

2.2. Two level segmentation

The valid syllable units can be determined using a duration analysis of syllables. [5] shows that a unit of length between 100ms and 300ms is a valid syllable. Similar analysis on manually segmented data of Tamil DDNews [8] gives an average syllable length of 133ms. A duration of range 56ms to 175ms is assumed to be valid. In the two level segmentation approach, the speech signal is segmented into large polysyllabic units using a coarse grain window and then according to the validity of duration, a second level of segmentation is performed using a fine grain window.

As stated in [5], the WSF value can be varied from 2 to 10. In the two level segmentation, the first phase of coarse segmentation is performed with a large WSF that will give a low resolution and hence polysyllabic or word boundaries are identified. WSF value of 5 or 6 is appropriate for detecting the word boundaries. Each of these polysyllabic units identified through duration analysis is segmented using WSF value of 4 or 3. The low WSF value will give better resolution and hence, detect the boundaries within a word.

An example for the boundaries detected by the two level segmentation is shown in Fig 1. The manual transcription of the sentence represented by the speech signal in this figure has 16 syllable like units. The baseline segmentation algorithm detected only 10 of them while, the two level segmentation extracted 15. The recognition performance for test data using single and two level segmentations are tabulated in Table 1. Recognition performance with WSF values of 4, 5 followed by 4 (5/4) and 6 followed by 3 (6/3) are shown. The two level segmentation gives better performance as the segmented syllable units are more accurate. In the next section, we discuss the algorithm for segmenting the text into syllable like units.

3. Text segmentation based on linguistic rules

The syllable can be defined as a vowel nucleus supported by consonants on either side, It can be generalized as a C*VC* unit where C is a consonant and V is a vowel. The linguistic rules to extract the syllables segments from a text is generated from spoken Tamil.

These rules can be generalized to any syllable centric language. The text is preprocessed to remove any punctuations. Numerals 1,2,3 and 4 in the algorithm represent the position of the alphabet in the text to be segmented. The basic rules are:

- Check the first character of the word:
- If the first character is a V then
  the second character is a C; Check the third character:
  - If the third character is a C; Check fourth character:
    - If 3^{rd} and 4^{th} characters are equal; VCC(123) is the syllable
    - else; VC (12) is the syllable
  - If the third character is a V; Then $V(1)$ is the syllable
- If the first character is a C then
  Check for second character:
  - Second character is a V
    The third character has to be a C; Check the 4th character:
    - If the fourth character is a C; Check for 5th character:
      5th character is a V; CVC (123) is the syllable
      5th character is a C or a word end; $CVCC$(1234) is the syllable
    - If the fourth character is a V; CV(12) is the syllable
  - Second character is a C; we assume that the 3rd character has to be a vowel, and subsequently the 4th character has to be a C
    Check for 5th character:
    - If the 5th character is a C or a word end; Then CCVC (1234) is the syllable
    - If the 5th character is a V; Then CCV(123) is the syllable

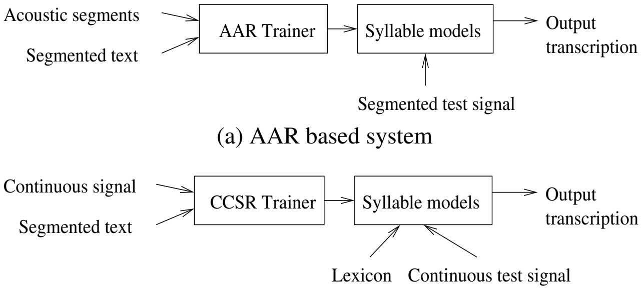
Text syllabification example using the above mentioned algorithm:

a	var |	sey	di	yA	Lar	ha	Li	tam
V	CVC |	CVC	CV	CV	CVC	CV	CV	CVC

’|’ represents a word boundary. It should be noted that each word is syllabified separately. After a syllable is identified from a word, the remaining part of the word is processed again by the algorithm. The text syllabification algorithm gives units comparable to the units given by group delay based segmentation. The two units can be made equivalent by using some specific language or domain rules. For example, it was observed that almost always ”di ru” was pronounced as a single unit ”diR”. Once the units are comparable or equivalent the segmented text can annotate the speech syllables. These syllabified texts can also be used to analyse syllable structure in the language like frequently occurring syllables or the syllables that can start or end a sentence. Syllable based N-gram language models can be built using these rules to segment large amount of text.

4. Experimental Setup

We use two different approaches to build a continuous speech recognizer, viz the automatically annotated continuous speech recognizer (AAR) and a conventional continuous speech recognizer (CCSR). Both these techniques illustrated in Fig 2 are explained below in detail.



(a) AAR based system

(b) CCSR based system

Figure 2: Two different Recognition systems

4.1. Automatically annotated recognizer (AAR)

Through out this paper, we assume that the sentence level transcriptions corresponding to the speech are available for the train data. The acoustic signal is segmented using the group delay based segmentation algorithm discussed in section 2. The corresponding text is syllabified using the rules explained in section 3. The text corresponding to the train data is annotated by matching both these segmented units. Once the train data is annotated, the syllable based continuous speech recognition system can be built. The unique syllables in the text are identified and the examples are extracted from the data according to the occurrence. As explained earlier [6] shows that speaking rate variations can be accounted by using MFS and MFR. This accounts for the spectral variations in the test speaker when compared to the train speakers. Using this scheme, multiple examples of a syllable can be extracted from a single utterance of the syllable. Thus, the scarcity of examples for a particular syllable is taken care. Models can be built for all unique syllables even if the occurrence rate is too low.

The block diagram in Fig 2(a) gives the details of AAR system. 1047 unique syllables are extracted from four Tamil DDNews bulletins [8] of 20 minutes duration each. The models are built using HMM based engine with 39 feature vectors which include 13 cepstral, velocity and acceleration components each. During testing, the test data is segmented using the group delay based segmentation and the syllables are recognized in isolated style. Each of the train and test data are of different female speakers.

4.2. Conventional continuous speech recognizer (CCSR)

The CCSR based technique is shown in Fig 2(b). In this approach, the segment boundary information of the acoustic signal is not used during training or testing. The syllabified text corresponding to speech signal in the train data is given as input to the recognizer. The training data for CCSR consists of continuous utterances and, in general, the boundaries dividing the segments of speech corresponding to each underlying sub-word (syllable) model in the sequence are not known. The HMM tool kit uses an *Embedded Training* mechanism followed by *Token Passing model* for state alignment. The earlier isolated style approach on the other hand uses a *Baum Welch Training* mechanism followed by *Viterbi* algorithm for state alignment. A syllabified dictionary and a list of possible syllable models are also given as input to the recognition engine.

The experiments are performed on the same train and test data. The transcriptions of the four Tamil news bulletins used for training are segmented into syllable like units using the text segmentation algorithm. These syllables form the basic syllable models to be built. The syllabified transcripts along with the acoustic signal are used as the train data. The same 39 feature vectors used for AAR are used in this approach.

4.3. MFS and MFR based feature extraction technique

Conventional speech processing systems window the speech signals into frames assuming the fact that the signal is quasi-stationary over a short period. The recognition systems built using single frame size and frame rate may cause problems when the test speaker's speaking rate is very different from that of speaker's data used during training. [6] explains in detail the MFS and MFR technique that captures the sudden changes in the spectral information. While the MFS takes care of the variations in the features of the acoustic signal, MFR takes into account the variations in the speaking rate of a speaker. As shown in [6], both these techniques can be combined to build a robust acoustic model from limited number of examples. Thus this method takes care of scarcity of examples for building a particular syllable model. Features are extracted from a single signal using multiple frame sizes varying from 12ms to 20ms and multiple frame rates by varying the frameshift from 10ms to 18ms.

4.4. Results

The AAR engine is tested using two different methods. The test data is a single news bulletin of 20 minutes duration not seen during training. This data is syllabified using the two level group delay based segmentation algorithm explained in section 2. The syllable units thus obtained are tested against the models generated using the MFS and MFR technique. The two testing techniques are the use of single frame size (SFS) and the use of MFS/MFR for the test data. The MFS/MFR technique uses a ranking scheme to extract the test result from the multiple instances of the same utterance. The results tabulated in Table 2, compares AAR with the CCSR engine. The AAR system could not be trained using SFS as syllables are trained in isolated style and not enough examples are available for all syllables. The CCSR is also tested using the two methods on the same test data. CCSR uses the MFS/MFR technique only during training as no voting scheme can be used for the test output. When train data is doubled, AAR gives a best performance of 52.02% when compared to 35.9% using CCSR.

Table 2: Recognition performance with different schemes

Training scheme	Test scheme		Test scheme	
	AAR		CCSR	
SFS	SFS	-	SFS	33.42%
	MFS	-	MFS	-
MFS	SFS	39.36%	SFS	21.17%
	MFS	44.52%	MFS	-

5. Discussions

It is easily seen that AAR gives better performance than CCSR. As the MFS and MFR based features are used in building the acoustic models, even if transcription is available for a smaller amount of train data, a reliable recognition system can be built. This system can be later on used for bootstrapping or clustering the untranscribed train data. As mentioned in [6] the MFS/MFR gives better results only if used during both testing and training. This accounts for results of SFS and MFS/MFR for CCSR We were unable to apply the MFS/MFR during testing for CCSR. The CCSR in general gives poorer results for the same train data because the boundary information is not available. The results shown in this paper are certainly better than the earlier work in [3] as models are build for all the syllables available and no pruning is done on basis of number of examples available.

The accuracy in the syllable boundaries given by the two level segmentation algorithm explained in section 2 can be attributed to the fact that the prosody varies considerably across the utterances. In the proposed technique, the following assumption is made: Prosody may vary substantially across words but might be rather uniform within a word. The first level segmentation gives the gross boundaries corresponding to large amplitude and large duration polysyllabic segments. The second level segmentation, then identifies the monosyllable boundaries within each polysyllable segment. The text segmentation rules are mainly linguistic rules and can be enhanced or altered according to the domain or type of speakers. These rules though developed using Tamil, can be easily extended for any syllable timed language.

6. Conclusions

This paper talked about two different types of continuous speech recognition engines, It is shown that a syllable based continuous speech recognition system can be built using automatically annotated train data. This new approach is used to build a reliable recognizer even when a small amount of transcribed unannotated train data is available. We have presented a novel idea to get better syllable boundaries from the group delay based automatic segmentation. We also discussed the linguistic rules for syllabifying syllable based languages. The recognition performance is shown to be better than a conventional HMM based continuous speech recognizer due to the reduction in search space complexity.

7. References

[1] Lamel, L., Jean-Luc Gauvain and Gilles Adda, ”Unsupervised acoustic model training”, Proceedings of IEEE Int. Conf. Acoust. Speech, and Signal Processing, Vol. 1, pp. 877-880, 2002.

[2] Steven Greenberg, ”Speaking in short hand - A syllable centric perspective for understanding pronounciation variation”, Speech Communications, Vol. 29, pp159-176, 1999.

[3] G. L. Sarada, N. Hemalatha, T. Nagarajan, and Hema A. Murthy, ”Automatic transcription of continuous speech using unsupervised and incremental training”, INTERSPEECH - 2004, pp405-408, Korea.

[4] V. Kamakshi Prasad., T. Nagarajan., Hema A. Murthy., ”Automatic segmentation of continuous speech using minimum phase group delay functions”, Speech Communications, Vol 42, pp429-446, 2004.

[5] T Nagarajan and Hema A Murthy, ”Subband-Based Group Delay Segmentation of Spontaneous Speech into Syllable-Like Units”, EURASIP, Volume 2004, No 17, pp2614-2625, 2004.

[6] Sarada, G. L., Nagarajan, T., Hema A. Murthy., ”Multiple Frame Size And Multiple Frame Rate Feature Extraction For Speech Recognition”, SPCOM-2004, December.

[7] T Nagarajan and Rajesh M Hegde and Hema A Murthy, ”Segmentation of Speech into Syllable-Like Units”, EUROSPEECH ’04, pp2893-2896, 2003.

[8] ”Database for Indian languages”, Speech and Vision Lab, IIT Madras, 2001.