Sādhanā Vol. 34, Part 2, April 2009, pp. 221–233. © Printed in India

Automatic transcription of continuous speech into syllable-like units for Indian languages

G LAKSHMI SARADA[1], A LAKSHMI[1], HEMA A MURTHY[1] and T NAGARAJAN[2]

[1]DON Laboratory, Computer Science Department, Indian Institute of Technology Madras, Chennai 600 036
[2]Department of Information Technology, SSN College of Engineering, Chennai 603 110
e-mail: {lakshmi,hema}@lantana.tenet.res.in; nagarajant@ssn.edu.in

MS received 20 April 2007; revised 28 June 2008

Abstract. The focus of this paper is to automatically segment and label continuous speech signal into syllable-like units for Indian languages. In this approach, the continuous speech signal is first automatically segmented into syllable-like units using group delay based algorithm. Similar syllable segments are then grouped together using an unsupervised and incremental training (UIT) technique. Isolated style HMM models are generated for each of the clusters during training. During testing, the speech signal is segmented into syllable-like units which are then tested against the HMMs obtained during training. This results in a syllable recognition performance of 42·6% and 39·94% for Tamil and Telugu. A new feature extraction technique that uses features extracted from multiple frame sizes and frame rates during both training and testing is explored for the syllable recognition task. This results in a recognition performance of 48·7% and 45·36%, for Tamil and Telugu respectively. The performance of segmentation followed by labelling is superior to that of a flat start syllable recogniser (27·8% and 28·8% for Tamil and Telugu respectively).

Keywords. Speech segmentation; unsupervised training; clustering algorithms.

1. Introduction

Over the last decade, speech recognition systems have come to be increasingly used in automated systems with spoken language interfaces. Transcription of a continuous speech signal into a sequence of words is a difficult task, as continuous speech does not have any natural pauses in between words. The conventional method of building a large vocabulary speech recogniser for any language uses a top-down approach to speech recognition (Huang & Acero 1993). What we mean by top-down is that these recognisers first hypothesize the sentence, then the words that make up the sentence and ultimately the sub-word units that make up the words. This requires a large speech corpus with sentence or phoneme level transcription of the speech utterances (Thomas Hain *et al* 2005; Ravishankar 1996). The transcriptions must also include dysfluent speech in order that the recogniser can build models for all the sounds present. In addition, it requires a dictionary with the phonemic/sub-word unit transcription of the words and extensive language models to perform large vocabulary continuous speech recognition. The recogniser outputs words that exist in the dictionary. When the recogniser is to be adapted to a new task or new language, an existing recogniser available for a particular language is used, but once again requires the building of a dictionary and extensive language models for the new language.

Building such a corpus, is a labour-intensive and time consuming process. In a country like India, that has 22 official and a number of unofficial languages, building huge text and speech databases is a difficult task. Most speech recognition systems use phoneme level transcriptions. More recently (Greenberg 1999; Ganapathiraju *et al* 2001), have shown that syllable based systems outperform comparable triphone based systems. As Indian languages are syllable-centred, the focus of this paper is to obtain a vocabulary independent syllable-level transcription of the spoken utterance. Once syllable-like units are obtained task/domain specific language models can be employed to build speech recognition systems. In the following, we review some of the conventional approaches to transcribe speech into its sub-word units without the use of dictionaries or language models.

Researchers have tried several ways to automate speech transcription, without compromising the accuracy of models trained from untranscribed data. Some of the commonly used techniques for speech transcription are briefly explained here.

- Rabiner *et al* (1982), have proposed a new method called bootstrapping for speech recognition. The amount of training data, that is required to transcribe large amount of new data, is increased using bootstrapping.
- Ljolje & Riley (1991) have used an automatic approach to segmentation and labelling of speech when only the orthographic transcription of speech is available.
- Kemp & Waibel (1998) have proposed a method for unsupervised training of the speech recogniser for TV broadcasts. In this procedure, a bootstrap recogniser is used to generate transcripts of untranscribed training material.
- Wessel & Ney (2001) have proposed an approach in which a low-cost recogniser trained with one hour of manually transcribed speech is used to recognize 72 hours of unrestricted acoustic data. These transcriptions are then used to train an improved recogniser.
- Lamel *et al* (2002) have shown that the acoustic models can be initialised using as little as 10 minutes of manually annotated data.
- Asela Gunawardana & Alex Acero (2003) have proposed an unsupervised adaption of acoustic models to a domain with mismatched acoustic conditions.

The basic idea behind all the above mentioned efforts is, to either (a) Use an existing speech recogniser to transcribe large amount of untranscribed data or (b) Use some amount of manually transcribed data to transcribe new data.

A few methods that do not require any manually annotated speech corpora for speech recognition are given below.

- Incremental maximum a posteriori estimation of HMMs is proposed by Gotoh & Hochberg (1995). This algorithm randomly selects a sub-set of data from the training set, updated the model using maximum a posteriori estimation, and this process is iterated until convergence occurred.
- Chang *et al* (2000) have proposed a new method in which articulatory-acoustic phonetic features are extracted from each frame of the speech signal and classification of phone is done by special purpose neural networks.

Given that Indian languages are syllable centered, the focus of this work is to segment the continuous speech signal into syllable-like units and then perform an isolated style recognition of the syllable-like units. A novel approach is explored for automatically segmenting and transcribing the continuous speech signal without the use of manually segmented and labelled speech corpora (Nagarajan & Murthy 2004). This approach consists of two phases: automatic segmentation followed by automatic labelling. In this approach, the continuous speech signal is first automatically segmented into syllable-like units using a group delay based segmentation algorithm. For completeness, we review this process of segmentation in Section 2. The segmented data is then clustered. The clustering process consists of two phases namely: (i) initial cluster selection and (ii) incremental training. The clustered segments are then manually labelled after listening. Individual HMM models are trained for each of the syllable-like units. During testing, the speech utterance is first segmented into syllable-like units. The segmented units are tested in an isolated style using the individual HMM models obtained during training. This is explained in section 3. In section 4, a new feature extraction technique that combines both multiple frame rate and multiple frame size is explored. In section 5, the syllable-based recogniser is also compared with conventional flat-start HMM-based syllable recogniser. In the flat-start recogniser syllable-level transcriptions of sentences are provided at the time of training. We do not use language models or dictionaries in the flat-start recognition system, since the focus of this paper is ONLY on syllable recognition.

2. Automatic segmentation of speech

Segmentation is a process of decomposing the speech signal into a set of basic phonetic units. The basic phonetic unit can be a phoneme or a syllable, based on the language. Several speech recognition systems consider syllable as a basic unit because of its better representational and durational stability relative to the phoneme (Wu *et al* 1998; Greenberg 1999; Ganapathiraju *et al* 2001). Osamu Fujimura (1975) has argued that the syllable will serve as the effective minimal unit in the time-domain. Since Indian languages are syllable-timed languages, syllable is considered as the basic sound-unit in this work. Prasad (2002) has demonstrated that manual segmentation at syllable-like units followed by isolated style recognition of continuous speech results in 65% recognition of syllable-like units.

In Prasad *et al* (2004), a method for segmenting the acoustic signal into syllable-like units is proposed. Several refinements are made to this technique in Nagarajan *et al* (2003) and segmentation performance is improved. In this paper, we consider the sub-band and group delay based segmentation algorithm, discussed in (Nagarajan *et al* 2003), for segmenting the continuous speech signal into syllable-like units. In this approach, long silences are first removed from the speech signal using a coarse energy based threshold. The resultant speech signal is then passed through a bank of three filters namely: all-pass, low-pass and band-pass. The group delay function of the output of each of these three filters is computed. In the group delay function the poles (peaks) correspond to syllable boundaries in the speech signal. The boundaries derived from these three different group delay functions are combined to get the final syllable boundaries.

Though this segmentation approach gives quite accurate results (about 80% of the boundaries are detected) there exist errors in syllable boundaries. Due to these errors, the syllable training process, in (Nagarajan & Murthy 2004) is found to be error prone.

2.1 *Issues due to the automatic segmentation algorithm and their remedies*

Major issues due to the segmentation algorithm are discussed in this Section. Several refinements are also made to the syllable segments in order to overcome these issues.

2.1a *Unusual duration syllables:* When a speech signal with a large number of syllable units is automatically segmented, few segments may get merged because of the strong co-articulation effect between the syllable units causing unusual duration syllables. The speech signal may therefore not be segmented exactly at the syllable boundaries, causing some segments to have more than one syllable (poly-syllable) or a fragment of a syllable. Syllable training is observed to be poor due to these merged syllables and syllable fragments. For example, the merged syllable/$seya$/ has two vowels (*V*) and two consonants (*C*). During unsupervised syllable training, this may therefore be clustered with another syllable based on the similarity in either *V* or *C* part of that syllable.

The problem due to these merged syllables and syllable fragments can be overcome by performing duration analysis on the syllable inventory. To illustrate this, duration analysis is carried out on four DD News bulletins (DDNews 2001), in which each bulletin corresponds to a different female speaker. The duration of each bulletin is about 15 minutes. Average duration of each sentence is observed to be about 2·5 seconds. The speech signals are automatically segmented into S syllable segments. Duration of these syllable segments is then analysed. The duration analysis results show that duration of approximately 95% of the syllables varies from 110 ms to 270 ms. The average duration of a syllable is observed to be 135 ms.

After the raw segmentation is obtained using group delay, we process the segments based on duration of the segment. If the syllable duration is either below 110 ms or above 270 ms, that particular syllable segment is removed. This is carried out during training phase only. This process removes approximately 5% of the segments. Removing 5% of syllable-like segments with unusual duration during training phase is affordable and it does not lead to any problem as the syllable models are trained in isolation and not directly from the continuous speech signal. This results in M number of syllables, where $M < S$.

2.1b *Silence segments at the syllable boundaries:* Presence of silence at the boundaries of syllables may result in poor syllable recognition. It is noticed that while generating HMMs, a syllable that has a small silence portion at the beginning or end, results in a HMM with a separate state for silence. Since the spectral characteristics of silence are almost the same for all syllables, presence of silence sometimes dominates the syllable recognition process. As a result, syllables that have very similar silence characteristics may get clustered together.

The drawback in syllable clustering due to the presence of a short duration silence at the boundaries of syllables can be overcome by prefixing and suffixing a silence segment of approximately 20 ms to each syllable segment irrespective of whether a particular segment already has a silence or not. If a syllable segment already has a silence, prefixing and suffixing silence to that segment merely increases the duration of silence and it will not have any adverse effects during testing. This procedure is termed as the silence normalisation (SN) procedure.

An experiment is performed to illustrate the effect of the silence normalisation procedure. For this experiment 10 syllable segments that consist of 10 examples in each class are considered. These syllable segments are prefixed and suffixed with a silence segment of *about 20 ms* and they are manually grouped into clusters based on the similarity of syllable segments. Models with 5-states and 1-Gaussian mixture/state are initialised for these clusters. To illustrate the syllable recognition performance with the silence normalisation technique,

Table 1. Isolated syllable recognition performance for the syllable *ni* before and after silence normalisation.

Syllable	Before SN	After SN
/ni/	26·58	48·10
/ki/	41·7	73·6

the following experiment is performed. Models were trained with and without silence normalisation. The recognition performance for two syllables /*ki*/and/*ni*/is shown in table 1.

As a result of the silence normalisation technique, the syllable $/ni/$ is properly clustered with a different syllable from the same class. From table 1, it can be observed that the silence normalisation procedure can give better syllable recognition performance compared to that without silence normalisation. In this paper, HMM models are built for silence normalised syllable segments. During testing, the syllables are first segmented using the group delay based approach, silence normalised and then tested against the syllable models. Since the recognition performance of the system is calculated only from the number of syllable segments recognised correctly, all the syllable segments, irrespective of their duration, are considered for testing.

3. Labelling of speech

The next stage in building a speech recogniser is to build acoustic models for the speech segments. This requires grouping of similar sounding units to get unique models for each sound unit. In this paper, an unsupervised and incremental training algorithm (Nagarajan & Murthy 2004) is explored for grouping similar sounding units. This mainly comprises of two phases, namely: (i) initial cluster selection and (ii) incremental training.

3.1 *Initial cluster selection technique*

Automatic segmentation of the continuous speech signal gives S $(s_1, s_2, \ldots, s_S)$ number of syllable segments. These syllable segments are subjected to duration analysis that results in M number of syllable segments, where $M < S$. Silence normalisation is carried out on all the M syllable segments. All the M syllable segments are used in the initial cluster selection procedure.

(i) The feature vector includes 13 dimensional MFCC, 13 dimensional velocity and 13 dimensional acceleration parameters extracted from each of these M syllable segments with multiple frame sizes (MFS) of 12, 14, 16, 18, and 20 ms. The reason for using MFS feature extraction is, when a single example is considered and single frame size (SFS) features are used to build a HMM with more than one Gaussian mixture/state[1], the variance of these mixtures may become zero. There are several ways of increasing the variance and adjusting the mean value such as introducing some additive noise to the same example and considering it as a different example of the same class. In our case, instead of introducing some unknown noise, we consider features extracted from multiple

[1] More mixtures are required to capture speaker variations

frame sizes, thus increasing the number of examples for the corresponding class. In this approach, using a frame size, let us say F_1, MFCC features are extracted from a single example. The same example is considered again and features are once again extracted using a different frame size (say F_2). This process is repeated for several frame sizes (12, 14, 16, 18, and 20 ms) and the features extracted from these different frame sizes are given as input to the model initialisation process. This procedure is carried out for all the M syllables.

(ii) For each of the syllables, using the feature vectors derived using multiple frame sizes, a separate HMM ($\lambda_1, \lambda_2, \ldots, \lambda_M$) is initialised. This leads to M syllable models.

(iii) The M syllable segments are recognised against all the available HMMs and 2-best recognition results are obtained resulting in P_i pairs of syllable models, where i corresponds to the number of the iteration. Each pair consists of two syllable models (say λ_i and λ_j) and each syllable model in a pair is a representative of 2^{i-1} syllable segments (cluster). Hereafter, the terminologies model and cluster are interchangeably used.

(iv) The P_i pairs of syllable clusters (models) are then pruned. The syllable cluster pruning process is explained below: As mentioned above, the recognition of each syllable segment results in a pair (two syllable models) and if one of members of a pair is repeated for any of the other syllable segments, then the corresponding pair is removed. The pairs that occur only once in the 2-best results are retained. This procedure is carried out using the 2-best recognition results of all the M syllable segments. This step will reduce the number of pairs.

(v) New models are trained using these reduced number of pairs. Since each member of a pair, after i-th iteration, will have $2^{(i-1)}$ syllable segments, each model now is trained with 2^i syllable segments.

Steps 3–5 are repeated m times, resulting in at least 2^m syllable segments in each cluster. The initial cluster selection procedure leads to N clusters ($C_1, C_2, \ldots, C_N$).

3.2 *Incremental training*

After selecting the N initial clusters ($C_1, C_2, \ldots, C_N$), where models are only initialised, parameters of the models of each of the clusters are re-estimated incrementally using the following steps. This training procedure is referred to as incremental training.

(1) Model parameters of the initial clusters ($C_1, C_2, ..., C_N$) derived from the initial cluster selection procedure are re-estimated using Baum–Welch re-estimation algorithm.

(2) These new models are used to decode all the syllable segments using Viterbi algorithm. Clustering is performed based on the decoded output.

(3) If a particular cluster is found to have less than k ($k = 3$) number of syllable segments, that particular cluster is removed and the number of models is reduced to N_0, where $N_0 < N$.

(4) Steps 1–3 are repeated until the convergence criterion is satisfied. The convergence criterion followed is explained in step 5.

(5) In each iteration, as the model parameters are re-estimated and the syllable segments are re-clustered, the number of syllable segments that migrate from one cluster to another is expected to reduce. The convergence is said to be met, if the number of syllable migrations between clusters reaches zero and the incremental training procedure is terminated.

This entire technique, including initial cluster selection procedure, is named as the unsupervised and incremental training (UIT) technique. Once convergence is met, a set of

Table 2. Syllable recognition performance of the UIT system.

	Performance of UIT system (in %)	
Sound units	Speaker dependent data	Speaker independent data
Complete syllable	56·2	42·6
CV/VC only	25·6	20·8

$S_1, S_2, \ldots, S_{N_0}$ HMMs is obtained. Each HMM represents a single syllable. These models are then labelled according to the syllable identity in the given language. This is done by manually listening to segments in each cluster. Clusters for which a definite identity can not be given are removed.

3.3 *Performance analysis*

The Indian television news bulletins of Tamil (DDNews 2001) are used to analyse the performance of UIT algorithm for transcribing speech data not seen during training. Training is performed using four female speakers' news bulletins, each of about 15 minutes duration and for testing, two female speakers' news bulletins are used.

The speech signals are automatically segmented into syllable-like units that results in S ($S = 8804$) syllables. Duration analysis and silence normalisation with 20 ms of silence is carried out on these syllable segments and M (M is about 8400) syllables are obtained. Using these syllable segments, the incremental training procedure is carried out. This results in N_0 number of HMMs, where $N_0 < M$. Each HMM is a 5-state and 2-mixture/state model. The total number of syllable clusters obtained after incremental training is 512. The final clusters are labelled manually, and the corresponding models with labels assigned to them are used for the syllable recognition task.

The data set considered for testing is divided into two categories namely (i) Speaker Dependent (SD) data and (ii) Speaker Independent (SI) data. In speaker dependent transcription, new data of a speaker used in training is transcribed. In speaker independent transcription, data of speaker that does not appear in training is transcribed. The performance of the syllable recognition system is computed as given below:

$$\frac{\text{No. of syllables correctly recognised}}{\text{Total no. of syllables}}.$$

The recognition results for Tamil language are given in table 2. The syllables which are correctly recognised are named as *Complete syllable* in table 2. In some cases, only the *CV* or *VC* part of the entire *CVC* syllable has been recognised correctly and they are categorised as *CV/VC only* in table 2. As a syllable recogniser, the UIT technique gives a syllable recognition performance of 56·2% and 42·6% for SD and SI data sets, respectively.

4. UIT using MFS and MFR cepstral features for syllable recognition

In the UIT technique, discussed in Section 3, initially a single example is considered for each class during initial cluster selection procedure and features are extracted with multiple frame sizes to initialise HMMs (refer Section 3.1). It is observed that most of the syllable clusters obtained in the initial cluster selection procedure itself, consists of syllable segments that are different examples of the same syllable.

Vaseghi *et al* (1997), claimed that, with the use of multiple frame size (multi-resolution) cepstral features, the localised features of the time-frequency trajectory of speech can provide crucial clues for classification. In addition to MFS feature extraction, several techniques have been proposed in the literature (Macias–Guarasa *et al* 2003; Samudravijaya 2004; Philippe Le Cerf & Dirk Van Compernolle 1994) that consider features extracted using different frame rates for speech recognition. In (Sarada *et al* 2004), it has been shown that the multiple frame size (MFS) and multiple frame rate (MFR) feature extraction technique can improve the recognition performance of an isolated style recogniser, if they are used both during training and recognition phases. In this paper, an attempt is made to make use of MFS and MFR cepstral features in UIT for syllable recognition task. The significance of multiple frame size and multiple frame rate features both during training and recognition of a syllable recogniser using UIT technique is discussed in this Section.

First the continuous speech signal is automatically segmented into *S* syllable-like units. The syllable segments are pruned using duration analysis and subjected to silence normalisation technique. The resultant M syllable inventory is used for the training process using MFS and MFR cepstral features. The initial cluster selection followed by incremental training procedure with MFS and MFR cepstral features is discussed in the following Sections.

4.1 *MFS and MFR based initial cluster selection*

The initial cluster selection process is the same as that of UIT (Section 3.1). The differences are: (a) instead of deriving features using multiple frame sizes alone, here, the cepstral features are extracted from a syllable segment using multiple frame sizes and frame rates and (b) during recognition also, instead of using a single feature vector (derived using 20 ms frame size) features are extracted with multiple frame sizes and frame rates from the same syllable segment and with each of these features separate recognition is carried out and combined, as explained below.

All M silence normalised syllable segments are considered and 39-dimensional cepstral feature vectors are extracted for the M silence normalised segments with multiple frame sizes (12, 14, 16, 18 and 20 ms) and frame rates corresponding to a frame shift of 10, 12, 14, 16 and 18 ms. These features are used to initialise the HMMs. The M syllable segments are then tested against all M HMMs. During recognition also, speech features are extracted using different frame sizes and frame rates. For each frame size and frame rate based features, a separate recognition experiment is conducted and 3-best results are considered for each syllable segment. These recognition results are combined using the decision combination rule discussed below:

For the 3-best results, weights are assigned based on their n-best position (1·0 for the 1st best, 0·8 for the 2nd best and 0·6 for the 3rd best). The score for each syllable model is computed based on its recognition position for different frame sizes and frame rates. An example of the recognition result of a syllable (say S_i) using MFS and MFR features is shown in table 3.

The entries in the first column correspond to a specific frame size/frame rate. In the table, S_{25}, S_{14}, and S_{35} represent the 25th, 14th, and 35th syllable models respectively. For a given syllable S_i, the 3-best recognition results are obtained using a combination of frame size and frame rate. In the table, F_1, F_2, and F_3 correspond to three different experiments. The 3-best results are ordered and weights are assigned as indicated in table 3.

Table 3. 3-best recognition results of a syllable using 3 different frame sizes and frame rates.

	3-BEST recognition results		
Frame size or frame rate	1st-best	2nd-best	3rd-best
F_1	$S_{25}(1{\cdot}0)$	$S_{14}(0{\cdot}8)$	$S_{35}(0{\cdot}6)$
F_2	$S_{25}(1{\cdot}0)$	$S_{35}(0{\cdot}8)$	$S_{14}(0{\cdot}6)$
F_3	$S_{14}(1{\cdot}0)$	$S_{25}(0{\cdot}8)$	$S_{35}(0{\cdot}6)$

To determine the identity of the syllable S_i, the following is performed:

The 3-best recognition results from all the recognisers are combined by adding the weights associated with the n-best position of the syllable. For example, S_{25} gets a weight of 2·8 (1·0 + 1·0 + 0·8), while S_{14} and S_{35} get a weight of 2·4 and 2·0 respectively. The unknown syllable S_i is thus identified as S_{25}. This recognition procedure is used to cluster the syllable segments, instead of relying on only one feature vector with fixed frame size as in the method presented in section 3. Except these two variations, the rest of the procedure is similar to the procedure presented in section 3. This modified initial cluster selection procedure results in N of syllable clusters.

4.2 *MFS and MFR based incremental training*

After selecting the initial clusters where the models are only initialised, they are re-estimated and subjected to the incremental training procedure. In Section 3.1, MFS feature extraction is used only during the initial cluster selection procedure. Since MFS and MFR feature extraction technique has shown a moderate improvement in the performance over that of SFS for an isolated style recogniser (Sarada *et al* 2004), this technique is adopted during incremental training for syllable recognition. The steps involved in the incremental training procedure are the same as in section 3.2. The incremental training procedure is carried out until the convergence criterion is satisfied. This procedure leads to N_0 syllable clusters and thus N_0 syllable models. A label is manually assigned to each of the syllable models based on a listening test. These syllable models with labels are further used for the recognition task using MFS and MFR cepstral features. The recognition task using MFS and MFR cepstral features is discussed in the following Section.

4.3 *MFS and MFR features for recognition*

The test speech signals are first segmented into syllable-like units and silence normalised. During recognition, features are extracted using multiple frame sizes and frame rates for the test syllables. For each frame size and frame rate, a separate recognition experiment is conducted and the results of the three best systems for each of the syllables are considered. The recognition results are combined using the decision combination rule explained in section 4.1 in order to get the final syllable recognition output.

5. Performance analysis using MFS and MFR based features

Performance of the syllable recognition system using UIT with MFS and MFR cepstral features is evaluated on the Indian language television news database (DDNews 2001). Two

Indian languages considered in this work are Tamil and Telugu. Section 5.1 analyses the syllable recognition performance on Tamil News bulletins, whereas Section 5.2 analyses the syllable recognition performance on Telugu News bulletins.

5.1 *Performance analysis on Tamil data*

The Tamil language news bulletins consisting of four female speakers are considered for training. The speech signals are automatically segmented into syllable-like units that result in S ($S = 8804$) syllables. Duration analysis and silence normalisation with 20 ms of silence is carried out on these syllable segments and M ($M = 8400$) syllables are obtained. Using these syllables, MFS (12–20 ms, in steps of 2 ms) and MFR (frame shift of 10, 12, 14, 16 and 18 ms) features are extracted and the modified incremental training procedure is carried out. Here each model is a 5-state and 2-Gaussian mixture/state model. This procedure results in 621 syllable models. If a syllable model consists of examples which are entirely different, that particular model is discarded (i.e. once the syllable models are obtained, they are manually pruned). A syllable identity is assigned to each syllable model after listening to the syllable sounds in the corresponding cluster. For recognition, two female speakers, which are not seen during training, are considered. During recognition, features are extracted using different frame sizes and frame rates for the test syllables. For each frame size and frame rate, a separate recognition experiment is conducted and 3-best results are combined to get the final recognition output.

We have also built a flat start HMM based recogniser using the syllable-level transcriptions. In this technique too, four female speakers data are considered for training. In the corpus considered, time-aligned syllable-level transcriptions are available. However, conventional syllable-based flat start training requires the speech signal and the corresponding syllable-level transcriptions ONLY. Without providing syllable boundaries, syllable models are trained using flat-start procedure. However, since most of the syllables do not have enough training examples, models for these syllables, cannot be trained properly. To overcome this, as in Ganapathiraju *et al* (2001), the transcriptions of those syllables can be replaced by their corresponding phoneme-level transcriptions. Since, in our work, our intention is to ONLY compare the performance of the syllable models trained using the proposed approach with that of the conventional flat-start training procedure, the unique syllable models trained using UIT-based approach alone are considered for both the techniques. The performance of the system is evaluated only for the 621 syllables that were obtained from UIT with MFS and MFR. Syllable recognition results using UIT with MFS and MFR feature extraction technique for the language Tamil is shown in table 4. These results are compared with the UIT technique with SFS cepstral features. The flat start HMM results are presented in column 4 of table 4. This clearly shows that for the given data segmentation followed by labelling results in a significant improvement in performance over flat start HMMs. It is important to note that flat-start recognisers do work well, when a few hours of data transcribed at the sentence level is available. As a syllable recogniser, the UIT with SFS gives a syllable recognition performance of 42·6% for test speakers not seen in training, whereas the UIT with MFS and MFR features gives a syllable recognition performance of 48·7%.

5.2 *Performance analysis on Telugu data*

For analysis purpose, the Telugu language news bulletins consisting of five male speakers are considered for training. The speech signals are again automatically segmented into syllable-like units that results in S ($S = 13773$) syllables. Duration analysis and silence normalisation with 20 ms of silence is carried out on these syllable segments and M ($M = 11885$) syllables

Table 4. Syllable recognition performance of UIT system using MFS, MFR features for training and testing for Tamil.

	Syllable recognition performance in %		
Syllables	UIT with SFS	UIT with MFS and MFR	Flat start Recognition
Complete syllable	42·6	48·7	27·8
CV/VC only	20·8	19·2	10·48

are obtained. Using these syllables, MFS (12–20 ms, in steps of 2 ms) and MFR (frame shift of 10, 12, 14, 16 and 18 ms) features are extracted and the modified incremental training procedure is carried out. Here each model is a 5-state and 2-Gaussian mixtures/state model. This procedure results in 485 syllable models. Once the syllable models are obtained, they are manually pruned. Again, a syllable identity is assigned to each syllable model after listening to the syllable sounds in the corresponding cluster.

For recognition, two male speakers, not seen during training, are considered. Using MFS and MFR cepstral features, syllable recognition is carried out. The syllable recognition results of UIT using MFS and MFR feature extraction technique for the language Telugu is shown in table 5. These results are compared with UIT technique with SFS cepstral features. The flat start HMM results are presented in column 4 of table 5. This again reinforces our conjecture that for the given data segmentation followed by labelling results in a significant improvement in performance over flat start HMMs. As a syllable recogniser, the UIT with SFS gives a syllable recognition performance of 39·94% for test speakers not seen in training, whereas UIT with MFS and MFR features gives a syllable recognition performance of 45·36%.

6. Conclusions and scope for future work

In this work, an attempt is made to automate the syllable transcription task for Indian languages that does not require any manually segmented and labelled speech corpus. In this work, a new feature extraction technique that uses multiple frame sizes and frame rates during both training and testing is explored to improve the syllable recognition performance.

The syllable recognition performance is evaluated for two Indian languages namely, Tamil and Telugu. A significant improvement over that of an analogous flat-start HMM based

Table 5. Syllable recognition performance of UIT system using MFS, MFR features for training and testing for Telugu.

	Syllable recognition performance in %		
Syllables	UIT with SFS	UIT with MFS and MFR	Flat start Recognition
Complete syllable	39·94	45·36	28·8
CV/VC only	14·05	14·06	9·64

recogniser is observed. The proposed approaches to syllable recognition have been applied to the Tamil and Telugu language corpus. In future it can be applied to other corpora for various Indian languages.

References

Asela Gunawardana, Alex Acero 2003 Adapting acoustic models to new domains and conditions using untranscribed data. In *In European Conference on Speech Communication and Technology.* Geneva, 1633–1636

Chang S, Sastri L, Greenberg S 2000 Automatic phonetic transcription of sponta- neous speech. In *Proceedings of Int. Conf. Spoken Language Processing* 4: 330–333

DDNews 2001 *Database for Indian languages.* India, Speech and Vision Lab, IIT Madras, Chennai

Ganapathiraju A, Hamaker J, Picone J, Ordowski M, Doddington G R 2001 Syllable based large vocabulary continuous speech recognition. In *IEEE Trans. Speech, Audio Processing.* 9: 358–366

Gotoh Y and Hochberg M M 1995 Incremental map estimation of hmms for efficient training and improved performance. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing* 877–880

Greenberg S 1999 Speaking in short hand - A syllable centric perspective for understanding pronounciation variation. In *Speech Communication* 29: 159–176

Huang, Acero 1993 *Spoken Language Processing.* (New Jersy: Prentice Hall)

Kemp T, Waibel A 1998 Unsupervised training of a speech recognizer using tv broadcasts. In *Proc. of ICSLP 98.* Vol. 5 Sydney, Australia, 2207–2210

Lamel L, Gauvain J-L, Adda G 2002 Unsupervised acoustic model training. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing.* 877–880

Ljolje A, Riley M D 1991 Automatic segmentation and labelling of speech. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing* 1: 473–476

Macias-Guarasa J, Ordonez J, Montero J M, Ferreiros J, Cordoba R, Haro L F D 2003 Revisiting scenarios and methods for variable frame rate analysis in automatic speech recognition. In *Proceedings of EUROSPEECH.* Geneva, 1809–1812

Nagarajan T, Murthy H A, Hegde R M 2003 Segmentation of speech into syllable-like units. In *Proceedings of EUROSPEECH.* 2893–2896

Nagarajan T, Murthy H A 2004 Non-bootstrap approach to segmentation and labelling of continuous speech. In *National Conference on Communication.* 508–512

Osamu Fujimura 1975 Syllable as a unit for speech recognition. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing.* 23: 82–87

Philippe Le Cerf, Dirk Van Compernolle 1994 A new variable frame rate analysis method for speech recognition. In *Signal Processing Letters IEEE.* Vol. 1. Greece, 185–187

Prasad V K 2002 Segmentation and recognition of continuous speech. Ph.D. thesis, Indian Institute of Technology, Department of Computer Science and Engg., Madras, India

Prasad V K, Nagarajan T, Murthy H A 2004 Automatic segmentation of continuous speech using minimum phase group delay functions. In *Speech Communication* 42: 1883–1886

Rabiner L R Rosenberg A E, Wilpon J G, Zampini T M 1982 A bootstrapping training technique for obtaining demisyllabic reference patterns. In *J. Acoust. Soc. Amer.* 71: 1588–1595

Ravishankar M K 1996 http://cmusphinx.sourceforge.net/sphinx3

Samudravijaya K 2004 Variable frame size analysis for speech recognition. In *Proceedings of Int. Conf. Natural Language Processing.* New Delhi, India 237–244

Sarada G L, Nagarajan T, Murthy H A 2004 Multiple frame size and multiple frame rate feature extraction for speech recognition. In *SPCOM.* Bangalore, India 592–595

Thomas Hain, *et al* 2005 Automatic transcription of conversational telephone speech. In *IEEE Trans. Speech, Audio Processing* 13: 1173–1185

Vaseghi S, Harte N, Milner B 1997 Multi-resolution phonetic/segmental features and models for hmm-based speech recognition. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing.* 2: 1263–1266

Wessel F, Ney H 2001 Unsupervised training of acoustic models for large vocabulary continuous speech recognition. In *IEEE Workshop on ASRU.* 307–310

Wu S D B E, Kingsbury, Morgan N, Greenberg S 1998 Incorporating information from syllable-length time scales into automatic speech recognition. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing.* 2: 721–724