Automatic Transcription of Continuous Speech using Unsupervised and Incremental Training

G.L.Sarada, N.Hemalatha, T.Nagarajan, Hema A.Murthy

Department of Computer Science and Engg.
Indian Institute of Technology, Madras
hema@lantana.iitm.ernet.in

Abstract

In [1], a novel approach is proposed for automatically segmenting and transcribing continuous speech signal without the use of manually annotated speech corpora. In this approach, the continuous speech signal is first automatically segmented into syllable-like units and similar syllable segments are grouped together using an unsupervised and incremental clustering technique. Separate models are generated for each cluster of syllable segments and labels are assigned to them. These syllable models are then used for recognition/transcription. Even though the results in [1] are quite promising, there are some problems in the clustering technique due to (i) the presence of silence segments at the beginning and end of syllable boundaries. (ii) fragmentation of syllables (iii) merging of syllables and (iv) poor initialization of syllable models. In this paper we specifically address these issues, make several refinements to the baseline system, which has resulted in a significant performance improvement of 8% over that of the baseline system described in [1].

1. Introduction

The last decade has witnessed substantial progress in speech recognition technology, with today's state-of-art systems being able to transcribe unrestricted broadcast news speech data with good accuracy. However, acoustic model development for these recognizers relies on the availability of large amounts of manually transcribed training data. Obtaining such data is both time-consuming and expensive, requiring trained human annotators and substantial amounts of supervision. To reduce the manual effort and the expenditure of transcribing speech data, the most commonly used approach is bootstrapping.

In [2], a low-cost recognizer trained with one hour of manually transcribed speech is used to recognize 72 hours of untranscribed data. These transcriptions are then used to train an improved recognizer. [3] uses an automatic approach to segmentation of labeled speech and labeling and segmentation of speech when only the orthographic transcription of speech is available. Lamel [4] has shown that the acoustic models can be initialized with as little as 10 minutes of manually annotated data. The basic idea behind all the above mentioned approaches, is to use an existing speech recognizer to transcribe huge amount of untranscribed data, which can further be used to refine the trained models.

The two basic problems in the above mentioned approaches are (i) if there is a mismatch in environment or language during transcription, the performance is expected to be very poor (ii) during refinement of the model parameters, convergence will be very slow and in some cases it may be impossible. The immediate alternative to this problem is, manually transcribing part of the new data, building models using this and then transcribe the rest of the data. For transcribing even a small amount of data, trained human annotators are needed. There is a need for a system to transcribe continuous speech signal without a manually annotated corpus. An automatic procedure for phonetic transcription of spontaneous speech has been developed in [5] which does not require transcription. In [5], articulatory-acoustic phonetic features are extracted from each frame of the speech signal and classification of phones is done by special purpose neural-networks. The output of these networks is processed by a Viterbi-like decoder to produce a sequence of phonetic-segment labels along with boundary demarcations associated with each segment.

In [1], a novel approach for automatic segmentation and transcription of speech data without using manually annotated speech corpora is proposed. The continuous speech signal is first automatically segmented into syllable-like units by considering the short-term energy as a magnitude spectrum of some arbitrary signal.Similar syllable segments are then grouped together using an unsupervised and incremental clustering technique. Separate models are generated for each cluster of syllable segments and labels are assigned manually. The syllable models of these clusters are then used to transcribe/recognize the continuous speech signal.

In [1], during incremental training, in some cases, clustering is poor due to syllable fragments and merged

syllables. For merged syllables, e.g /**vana**/, as it has two vowels and two consonants, because of the similarity in vowel/consonant part alone, it is clustering with some other syllable which also has similar vowel/consonant part. In some cases, because of the presence of syllable fragments, e.g /**k**/, as it has short duration segment which consists of consonant alone, it is clustered with some other syllable which has a similar consonant part. In some cases, because of the presence of silence at the boundaries of the syllables, the clustering is poor. The issue here is that different syllable segments which have silence at the beginning get clustered together owing to the first state in the HMM being assigned to silence. A similar clustering results for syllables with a silence portion at the end, as the last state is assigned to silence in the HMM. Further, if the initial models are seeded properly with seemingly unique syllables, significant improvement in the incremental clustering approach can be expected. In [1], the first 2000 syllables are only taken for initial cluster selection procedure. Here because of the limited data used, the models are not accurate after initial cluster selection procedure. So the clustering is poor.

In this paper, we will specifically address these issues and made some refinements to the base-line system [1] to overcome the above mentioned problems during incremental training.

The remainder of this paper is organized as follows. In section 2, we briefly discuss the group delay based segmentation procedure which is adopted to segment the speech signal into syllable-like units. Then we address the above mentioned issues and we describe the refinements that are made to the base-line system in [1] to improve the performance of the clustering technique. In section 3 the performance of this approach is analyzed and compared with the performance of the base-line system.

2. Automatic segmentation followed by labeling

For both training and testing, Indian television news bulletins of the language Tamil have been used [6]. During training, four speakers data, each of 15 minutes duration are used. During testing, two news bulletins of the language Tamil are used. The training and testing set speakers are different. Here, the total duration of the speech signal of each news bulletin is split into small segments of approximately 2.5s each.

2.1. Syllable-like segmentation

The syllable is structurally divisible into three parts the onset, nucleus, and coda [5]. Although many syllables contain all the three elements, say **CVC**, a significant portion contain one element typically, **V** or two elements **CV or VC**. In [7], a method for segmenting the acoustic signal into syllable-like units is proposed. Using this approach four speakers speech data are segmented into syllable-like units, which gives M syllable segments, s_1, s_2,..., $s_{\mathcal{M}}$ ($\mathcal{M} = 8800$).

During incremental training, in some cases, clustering is poor due to syllable fragmentation and merging. The problems due to syllable fragmentation and merging can be overcome as explained below. The durational analysis performed on the syllable inventory shows that the duration of $\approx 95\%$ of the syllables vary from 110ms to 270ms. The mean duration of syllable data is 135ms. If the syllable duration is below 110ms and above 270ms, that syllable segment is removed. By doing so, most of the syllable fragments and merged syllables are removed resulting in $\mathcal{N}$ syllable segments ($\mathcal{N} = 8400$).

In some cases, because of the segments having short silence portion at the boundaries of the syllables, the clustering is poor during incremental training. As initial and final states of HMMs are also equally important for recognition, if a syllable has a short silence portion at the boundary, while generating the models, a state will be assigned to silence. During clustering, the recognizer will try to recognize that syllable with some other similar but not identical syllable which also has silence portion at the boundaries. Because of this, wrong syllables are clustered together. This problem can be overcome by prepending and appending short duration silence ($\approx 20ms$) to each syllable segment. During incremental training, separate states are generated for both the silence portions. Since all syllables now have a silence segment at the boundaries, the clustering is accurate. After adding silence at the syllable boundaries, these syllable segments are used during the training process. The training process is similar to the training process in base-line system [1].

2.2. Initial cluster selection

For any iterative-training process, the assumed initial condition is crucial for the speed of convergence. After having all the segments, the initial groups of syllables are carefully selected to ensure fast convergence. In [1], during initial cluster selection procedure, $\mathcal{N}1$ syllable segments are selected from the $\mathcal{M}$ syllable segments where $\mathcal{N}1 < \mathcal{M}$. During initial cluster selection, if identical syllables are used to build models for each cluster, the incremental training procedure can lead to faster convergence. In [1], the first 2000 syllables are used, out of 8000 syllables, for initial selection procedure. Because of the limited data used during initial cluster selection, some clusters have identical syllable segments while other clusters have syllables which are similar in some sense. To ensure that initial clusters have identical segments, the following procedure is adopted.

1. All $\mathcal{N}$ ($\mathcal{N} = 8400$) syllable segments are taken for initialization.
2. Features (13 MFCC + 13 delta + 13 acceleration)

are extracted from these M syllable segments with multiple resolutions (i.e, with different window sizes and frame shifts). Multi-resolution feature extraction ensures a reasonable variance for each Gaussian mixture in the models.

3. M Hidden Markov Models (λ_{s1}, λ_{s2}, ..., $\lambda_{s\mathcal{N}}$) are initialized. To initialize model parameters, the Viterbi algorithm is used to find the most likely state sequence corresponding to each of the training examples, then the HMM parameters are estimated as in [1].
4. Using Viterbi decoding process, the same $\mathcal{N}$ syllable segments are decoded using 2-best criteria, resulting in $\mathcal{N}$ pairs of syllable segments(p_1, p_2, ..., $p_{\mathcal{N}}$) as in [1].
5. Among $\mathcal{N}$ pairs (p_1, p_2, ..., $p_{\mathcal{N}}$), if a syllable segment is found to be repeated in more than one pair, the other pairs are removed and thus the number of models are pruned.
6. New models are created with these reduced number of pairs.
7. Steps 4-6 are repeated for **m** times (here $m = 3$). After **m** iterations, each cluster will have 2^m syllable segments grouped together.

This initial cluster selection procedure will lead to $\mathcal{N}1$ clusters ($c_1, c_2, ..., c_{\mathcal{N}1}$) where $\mathcal{N}1<\mathcal{N}$. Due to this procedure almost all the clusters have identical syllable segments. In the next step the model parameters are re-estimated using a procedure which we refer to as incremental training.

2.3. Incremental training

After selecting the initial clusters (c_1, c_2, ..., $c_{\mathcal{N}1}$), where the models are only initialized , the parameters of the models of each of the clusters are re-estimated using Baum-Welch re-estimation procedure as in [1]. This training procedure is referred to as incremental training [1]. The steps followed in incremental training are given below.

1. The model parameters of the initial clusters (c_1, c_2,..., $c_{\mathcal{N}1}$) derived from the previous step are re-estimated using Baum-Welch re-estimation. Each model is a 7 state 1 Gaussian mixture HMMs.
2. The new models are used to decode all the syllable segments (s_1, s_2, ..., $s_{\mathcal{N}}$) using Viterbi decoding.
3. Clustering is done based on the decoded sequence.
4. If a particular cluster is found to have less than ϵ (Here, $\epsilon = 6$) syllable segments, that cluster is removed and number of models are reduced.
5. Steps 1-3 are repeated until convergence is met. The convergence criteria followed is similar to [1] and is explained below.

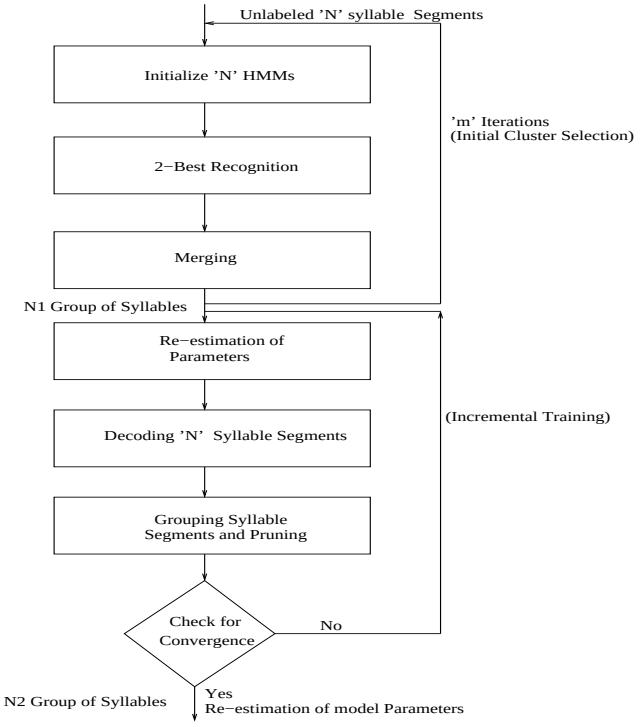

Figure 1: *Flow chart : Unsupervised and Incremental Training*

2.4. Convergence criteria

In each iteration as the model parameters are re-estimated and the syllable segments are re-clustered, the number of syllable segments which migrate from one cluster to another is expected to reduce at each iteration. The convergence criteria followed for the incremental training is based on **number of migrations between clusters** as in [1]. The convergence is said to be met if the number of migrations between clusters reaches zero. When this condition is met incremental training procedure terminates. This incremental training procedure will produce $\mathcal{N}2$ ($\mathcal{N}2 < \mathcal{N}1$) syllable clusters ($c_1$, c_2, ..., $c_{\mathcal{N}2}$), and in turn $\mathcal{N}2$ syllable models (λ_1, λ_2, ..., $\lambda_{\mathcal{N}2}$). After incremental training, almost all the syllable segments are found to be identical/similar in each cluster, with a few exceptions.

2.5. Labeling Clusters and Transcription

For using the above derived models for transcription/recognition tasks, it is required to assign a label for each of the clusters. By manually listening to the syllable segments in each of the clusters, a label as appropriate for the given sound is assigned as in [1]. Now the models are ready with labels assigned to them and they can be used for transcription/recognition of speech data. For performance analysis, the automatic transcription and manual transcription of a speech signal are shown (Fig.2).

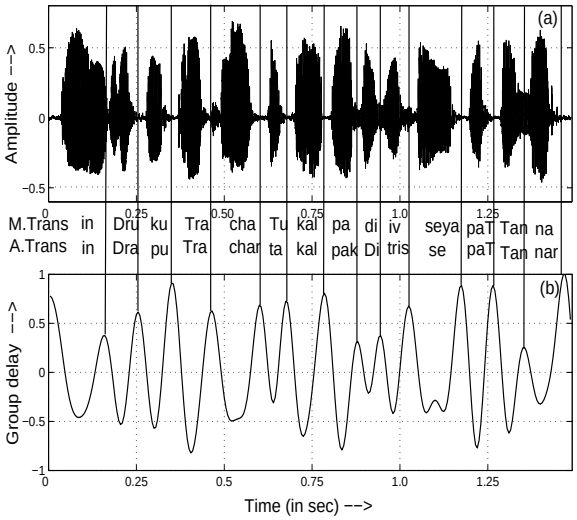

Figure 2: *Performance analysis - (a) an example of speech signal. (b) Group delay spectrum of the speech signal. A.Trans - Automatic transcription. M.Trans - Manual Transcription.*

In Fig.2b, the group delay function of the speech signal alone is shown. The peaks in Fig.2b correspond to the syllable boundaries of the speech signal which are obtained using the group delay based segmentation approach explained in [7].

3. Performance Analysis

To evaluate the performance, the system is trained on data from two speakers. During testing, two kinds of data are considered: (i) Untranscribed data corresponding to that of speaker used in training. (ii) Untranscribed data corresponding to that of speaker not used in training. They are referred to as I & II in Table.1. For II, after segmenting the data into syllable-like units, short duration silence of $\approx$ $20ms$ is prepended and appended to the syllable boundaries as is done for training data. In [1] as a syllable recognizer, for I and II, the average performance obtained was $\approx$ 42% and 35% respectively (Table.1). After refining the base-line system, for the same data as in [1], there is an improvement in the performance. For I and II data, the average performance is 56% and 42.6% respectively (Table.1). From Table 1, as a syllable recognizer, a significant performance improvement of 15% is observed for I and 8% for II. As a $\mathbf{CV/VC}$ unit recognizer, there is an improvement of 22% and 12% in the performance for I and II respectively. There is a considerable reduction in the performance for False case.

4. Conclusions

We have refined the base-line system in [1] to improve the performance of transcription, which segments and transcribes the continuous speech signal without the benefit of manually annotated speech corpus. We have shown the performance of 56% and 42% for known and unknown speaker data respectively. The results shows that, there is a performance improvement in the transcription system for both train data as well as test data, compared to the performance of the base-line system in [1]. The results are comparable in performance to the conventional batch training procedure, which uses manually annotated speech corpora.

Table 1: *Performance (in %) analysis of baseline system before refinement and after refinement*

Sound units	Before refinement		After refinement	
	I	II	I	II
Syllables	41.98	34.98	56.2	42.6
CV+VC	18.52	16.7	25.6	20.8
Vowel only	27.30	31.0	13	27.2
Cons. only	3.25	4.285	2.4	3
False	8.95	13.03	2.8	6.4

5. References

[1] Nagarajan., T. and Murthy., H. A., "An approach to segmentation and Labeling of continuous speech without bootstrapping", NCC-2004, pp.508-512, Jan 2004.

[2] Frank Wessel and Herman Ney, "Unsupervised training of acoustic models for large vocabulary continuous speech recognition", IEEE workshop on ASRU, pp.307-310, Dec 2001.

[3] Ljolje., A. and Riley., M. D., "Automatic segmentation and labeling of speech", ICASSP-1991,Vol.1, pp.473-476, April 1991.

[4] Lori Lamel, Jean-Luc Gauvain and Gilles Adda, "Unsupervised acoustic model training", ICASSP-2002, Vol.1, pp.877-880. May 2002.

[5] Shuangyu Chang, Lokendra Shastri and Steven Greenberg, "Automatic phonetic transcription of spontaneous speech, (American English)", ICSLP - 2000, Vol.4, pp.330-333.

[6] Database for Indian Languages, India, speech and vision laboratory, IIT Madras, Chennai-2001.

[7] Prasad., K. V., Nagarajan., T. and Murthy., H. A., "Automatic segmentation of continuous speech using minimum phase group delay functions, Speech Communications, Vol.42, pp.429-446, April 2004.