Automatic Transcription of Broadcast News

S. S. Chen, E. Eide, M. J. F. Gales, R. A. Gopinath, D. Kanvesky and P. Olsen
IBM T.J.Watson Research Center
Yorktown Heights, NY 10598
email: rameshg@us.ibm.com

March 7, 2001

Contents

1 Introduction **5**

2 Automatic Segmentation And Clustering **7**
2.1 Model Selection 7
2.2 Change Detection via BIC 8
2.2.1 Detecting One Change Point 8
2.2.2 Detecting Multiple Changing Points 9
2.2.3 Change Point Detection on 1997 Hub4 Evaluation Data 9
2.3 Clustering via BIC 10
2.3.1 Hierarchical Clustering via greedy BIC 10
2.3.2 Speaker Clustering on the Hub4 1996 evaluation data 11

3 Acoustic Modeling **11**
3.1 Bayesian Information Criterion for Gaussian Mixtures 11
3.2 Semi-tied Covariance Matrices 12
3.2.1 Training Semi-Tied Covariance Matrices 13
3.3 Pronunciation Networks 13
3.3.1 Generating Observations of Strings 14
3.3.2 Pronunciation Models given Observed Strings 15
3.3.3 Using Pronunciation Networks in Recognition 16
3.4 Tail Distribution Modeling 16
3.4.1 Richter Distributions 16
3.4.2 Power Exponentials 18
3.4.3 Tail-distribution Modeling - Results 20

4 Adaptive Training **20**
4.1 Modified Speaker Adaptive Training 21
4.1.1 Adaptation Transform 21
4.1.2 Canonical Model 21
4.2 Cluster Adaptive Training 22
4.2.1 Adaptation Transform 22
4.2.2 Canonical Model 22
4.3 Adaptive Training - Results 23

5 Language Modeling **24**

6 Some Additional Techniques **24**
6.1 Rover 24
6.2 Fast Speaking Rate 26

7 Summary **26**

8 Acknowledgements **27**

List of Figures

1 Detecting one changing point 9
2 Error analysis of change detection 10
3 Choosing the number of Gaussians by maximizing the BIC criterion 12
4 The distribution of powers, α, after training using (28) on the Hub4 1997 data. 19

List of Tables

1 Change detection error rates . . . 10
2 Segmentation and clustering in Hub4 1997 task . . . 10
3 MLLR adaptation enhanced by BIC clustering . . . 11
4 Comparison of the BIC approach with the thresholding approach on the 1997 evaluation subset. . . . 12
5 Semi-tied covariances: % WER on the 1996 Hub4 evaluation data a) Baseline b) Single semi-tied class c) Four semi-tied classes. . . . 14
6 Error rates on Broadcast News data. Condition A is the baseline. Condition B uses pronunciation networks. F0=Clean/Planned Speech. F1=Clean Spontaneous Speech. F2=Speech on Telephone Channels. F3=Speech with Background Music. F4=Speech in Degraded Acoustics. F5=Non-native speakers. FX=Combinations of F1-F5. . . . 16
7 Results on the Hub4 1997 partitioned evaluation test set . . . 20
8 Results for adapting the standard SI system on the Hub4 1997 partitioned evaluation test set . . . 23
9 Results for the MSAT system on the Hub4 1997 partitioned evaluation test set . . . 23
10 Results for CAT system on the Hub4 1997 partitioned evaluation test set . . . 24
11 Mixture LM and single LM . . . 25
12 Enumeration of individual systems used in the 1998 hub4 evaluation and some corresponding rover combinations. WER I and WER II refers to the error rates of the two individual test sets in the 1998 hub4 evaluation. . . . 25
13 Decoding results for M_0 and M_2 speech metrics . . . 26

Abstract

This paper describes the IBM approach to Broadcast News Transcription. Typical problems in the Broadcast News Transcription task are segmentation, clustering, acoustic modeling, language modeling and acoustic model adaptation. This paper presents new algorithms for each of these focus problems. Some key ideas include Bayesian Information Criterion (for segmentation, clustering and acoustic modeling) and Speaker/Cluster Adapted Training.

Abstract

Dieser Beitrag beschreibt ein bei IBM entwickeltes Verfahren zur Tranksription von Rundfunknachrichten. Zu den typischen Problemen der Transkription gehoeren die Segmentierung, Clusterung, akustische Modellierung, Sprachmodellierung und akustische Modelladaption. In disem Beitrag werden neuartige Algorithmen fuer diese einzelnen Probleme praesentiert. Als einige der hier enthaltenen Schluesselideen koennen das Kriterium der Bayes'schen Information (zur Segmentierung, Clusterung and akustischen Modellierung) sowie sprecher- und clusteradaptives Training genannt werden.

Abstract

Cet article décrit l'approche d'IBM pour la transcription des nouvelles du journal télévisé. Les problèmes caractéristiques de cette tâche sont la segmentation, la classification automatique, la modélisation acoustique, l'estimation de modèles de langage et l'adaptation des modèles acoustiques. Cet article présente de nouveaux algorithmes pour chacun de ces problèmes. Parmi les idées clés, on peut citer le critère d'information Bayesien (pour la segmentation, la classification automatique et la modélisation acoustique) et l'apprentissage adapté pour chaque locuteur/classe de locuteurs.

1 Introduction

Automatic transcription of Broadcast News (BN) is an extremely challenging and interesting large vocabulary continuous speech recognition (LVCSR) task. The dificulty lies in an intermingling of speech and non-speech events such as music and background noise. Moreover, the speakers have different speaking styles and accents, degraded by microphone or channel conditions and with background noise and sometimes background speech. Any solution to this problem, depending on transcription accuracy and speed, enables a wide range of useful applications such as close-captioning and audio-indexing of multimedia content.

The past few years have seen significant algorithmic advances in BN transcription in U.S English. This research effort has been sponsored by DARPA, through acoustic and textual data collections distributed by the Linguistic Data Consortium (LDC) and through yearly evaluations conducted by NIST. About 160 hours of acoustic training data and nearly 400 million words of textual training data in the BN domain is available from LDC.

Automatic transcription of BN is a multi-faceted problem that can not reasonably be expected to be resolved with a single algorithmic technique. The effort at IBM has spanned a whole spectrum of algorithmic techniques and innovations. This paper reviews techniques developed by the authors over the last few years and deployed by IBM in the 1997-9 Hub4 evaluations. The issues addressed span clustering and segmentation, acoustic modelling with and without channel adaptation, pronunciation modeling, language modeling as well as techniques to combine individual improvements (e.g. ROVER). While several of the techniques were developed in the context of BN transcription, they are general techniques that can be applied to other speech recognition tasks.

Segmentation of continuous audio into speaker/channel turns is useful in state-of-the-art speech recognition systems that employ speaker/channel adaptive schemes like cepstral mean normalization. Moreover, if the content is BN, one would like to *segment* the audio stream into homogeneous regions according to speaker identity, environmental condition and channel condition so that regions of different nature can be handled differently: for example, regions of pure music and noise can be rejected; also, one might design a separate recognition system for telephone speech. The various segmentation algorithms proposed in the literature fall into three categories: *decoder-guided* [30, 26], *model-based* [20] and *metric-based* [23, 24, 28]. The decoder-guided segmentation only places boundaries at silence locations, which in general has no direct connection with the acoustic changes in the data. Both the model-based and the metric-based segmentation schemes rely on thresholding of measurements which lack stability and robustness. More importantly, they do not generalize to unseen acoustic conditions. In contrast the so-called BIC segmentation algorithm described in this paper is robust, threshold-free and generalizes well to unseen acoustic conditions and is based on detecting *any* change in the acoustic condition.

When speaker turns have been identified, it is useful to adapt the acoustic models to a given speaker for improved recognition performance. Since the same speaker may occur several times in a BN clip, it is necessary to cluster all segments from the same speaker to get the maximum gain from adaptation. Typically hierarchical clustering schemes are used for this purpose [28, 26, 24]. In such schemes it is often difficult to determine the number of clusters. One can heuristically pre-determine the number of clusters or the minimum size of each cluster; accordingly, one can go down the tree to obtain desired clustering [30]. Another heuristic solution is to threshold the distance measures during the hierarchical process; the thresholding level is tuned on a training set [28]. Jin et al. [25] shed some light on automatically choosing a clustering solution. In this paper, we view clustering as a model selection problem and show that the BIC criterion is an effective termination criterion for hierarchical clustering.

The diversity of BN speech data (several speakers, channel, and noise conditions) implies large variance in the trained acoustic models. By classifying acoustic training and test data into homogeneous conditions we may build separate acoustic models for each condition. This paper proposes two adaptive training methods: modified Speaker Adaptive Training and Cluster Adaptive Training, both of which give significant performance improvements.

This paper also describes improvements to the modeling improvements that have a purely statistical basis. The classical modelling paradigm is expanded to include some modelling of covariance structure and alternatives to the gaussians density functions are suggested.

In speech recognition one models highly correlated features using diagonal Gaussians, even though full covariance Gaussians are more appropriate. Significant performance improvements are demonstrated using semi-tied covariances which model linearly transformed features using diagonal Gaussians.

A diagnostic study of histograms of any single dimension of the acoustic data assigned to a particular HMM state of a speech recognition system shows that these distributions are a) skew-symmetric, b) peakier around the mean and c) have fatter tails than a typical Gaussian distribution. This paper proposes two efficient alternatives to model (b) and (c) that avoids dramatically increasing the number of Gaussian mixture components - Richter distributions and power-exponential distributions. Marginal performance improvements are obtained using these distributions.

Speech recognition systems typically use a hand-generated lexicon. Inconsistency between the pronunciations in the lexicon and the way a speaker pronounces the words is a major source of degradation in performance. A proposed resolution of this problem make use of network topologies for phonemes that incorporate common deviations in pronunciation from the lexicon.

Another problem area related to pronunciation deviation is fast speech. Fast speech is well-known to significantly impair recognition accuracy [13]. This paper proposes two approaches - phone models with skip arcs, and faster-rate signal processing to handle fast speech.

In the Broadcast News domain there is a significant amount of training data. This paper shows that more accurate language models can be obtained by mixing language models built on subsets of the training corpus that span different periods in time rather than pooling all of the data and building a single model.

Schemes that combine hypothesis transcripts from several recognizers can greatly improve the accuracy. This is demonstrated using the ROVER algorithm [18] on a set of recognizers.

Our basic approach to BN transcription can be summarized as follows. First the continuous audio stream is segmented into acoustically homogeneous clips of audio. Typically the audio in each segment is from a single speaker with a fixed background noise level and channel condition. A first-pass recognition of these segments is done using single canonical acoustic model built on all the available training data using adaptive training. Adaptive training takes into account the wide range of variability in the acoustic training data and produces a sharp canonical model. The first-pass recognition is followed by adaptation of the canonical model to each segment using the recognition transcripts as ground truth. This is followed by a second-pass recognition. The process of adapting the acoustic models for each segment and refining the transcription by recognition with the adapted models is iterated several times. The quality of the adaptation is linked to the amount of adaptation data in each segment. Since a typical broadcast show has several segments from the same speaker, the amount of adaptation data is increased by clustering the segments into similarity groups. Acoustic model adaptation is done on the clusters rather than on individual segments.

The recognizer used in all the experiments in this paper is the standard IBM rank-based single-pass stack decoder (see [2, 3] for details). The training data used for the experiments is the complete LDC Hub4 Broadcast News distributions (both for acoustic and language models). Depending on when the experiments were conducted different test data sets are used in the results reported in the paper. However, they are all drawn from either the 1996, 1997 or 1998 Hub4 evaluation data sets (or subsets thereof). Results are reported using the classification of the speech data along the so-called F-conditions [35]: prepared speech (F0), spontaneous speech (F1), low fidelity speech, including telephone channel speech (F2), speech in the presence of background music (F3), speech in the presence of background noise (F4), speech from non-native speakers (F5) and FX - all other speech.

2 Automatic Segmentation And Clustering

The key idea is to view segmentation and clustering as model selection problems. This allows us to use the popular Bayesian Information Criterion (BIC) model selection criterion for segmentation and clustering. In segmentation the goal is to detect changes in speaker identity, environmental condition and channel condition without knowing the class of changes *a priori*. The paper proposes a maximum likelihood approach to detect turns of a Gaussian process; the decision of a turn is based on the *Bayesian Information Criterion* (BIC), a model selection criterion. The paper also proposes applying BIC as a termination criterion in the hierarchical methods for speaker clustering. Experimental results indicate that the segmentation algorithm successfully detects acoustic changes and the clustering algorithm can produce clusters with high purity and enhance un-supervised adaptation as much as the ideal clustering by the true speaker identities.

2.1 Model Selection

The problem of model selection is to choose one among a set of candidate parametric models to describe a given data set. Models with too few parameters will not be accurate (under-fitting), while those with too many parameters will not generalize (over-fitting). There are two popular methods for model selection: cross-validation (non-parametric) and Bayesian Information Criterion (parametric) [29].

Consider modeling the data $\mathcal{X} = \{x_i : i = 1, \cdots, N\}$ using one model from a set of parametric models $\mathcal{M} = \{M_i : i = 1, \cdots, K\}$. For any given model M assume that the $\#(M)$ parameters are chosen to maximize the likelihood and let $L(\mathcal{X}, M)$ denote this maximum value. BIC is a likelihood criterion penalized by the number of parameters in the model and is defined as

$$BIC(M) = \log L(\mathcal{X}, M) - \lambda \frac{1}{2} \#(M) \times \log(N) \tag{1}$$

where the penalty weight $\lambda = 1$. One chooses the model M with maximal BIC value. This procedure can be shown to be a large-sample version of the Bayesian procedure for the case of independent, identically distributed observations and linear models [29]. By varying the penalty weight $\lambda > 0$, one can tradeoff the relative importance of likelihood and model complexity, although only $\lambda = 1$ corresponds to the strict definition of BIC.

2.2 Change Detection via BIC

In this section, we describe a maximum likelihood (ML) approach for acoustic change detection based on BIC. The feature sequence $\mathbf{x} = \{x_i \in \mathcal{R}^d, i = 1, ..., N\}$ extracted from the continuous audio stream (e.g., mel-cepstral features) is assumed to be drawn from independent multivariate Gaussian process: $x_i \sim N(\mu_i, \Sigma_i)$, where μ_i and Σ_i are the mean and covariance respectively.

2.2.1 Detecting One Change Point

Consider a simplified problem with at most one change point. A change at time i is resolved with the following hypothesis test: $H_0 : x_1 \cdots x_N \sim N(\mu, \Sigma)$ versus $H_1 : x_1 \cdots x_i \sim N(\mu_1, \Sigma_1); x_{i+1} \cdots x_N \sim N(\mu_2, \Sigma_2)$. If Σ, Σ_1 and Σ_2 are the sample covariances from all the data, from $\{x_1, \cdots, x_i\}$ and from $\{x_{i+1}, \cdots, x_N\}$, respectively, then, the ML ratio statistics are

$$R(i) = N log|\Sigma| - N_1 log|\Sigma_1| - N_2 log|\Sigma_2| \tag{2}$$

and the ML estimate of the changing point is $\hat{t} = \arg\max_i R(i)$.

The above test can be viewed as a model selection problem problem. The data is modeled as one Gaussian or two Gaussians. The difference in BIC values of these two models is

$$BIC(i) = R(i) - \lambda P \tag{3}$$

where the penalty $P = \frac{1}{2}(d + \frac{1}{2}d(d+1)) \log N$, and the penalty weight $\lambda = 1$; d is the dimension of the space. If (3) is positive, the model with two Gaussians is favored. A change has occurred if $\{\max_i BIC(i)\} > 0$. Clearly the ML changing point can also be expressed as $\hat{t} = \arg\max_i BIC(i)$.

Compared with the metric-based segmentation methods ([28, 23]) the proposed scheme is robust. The main reason appears to be that these schemes use the distance between *fixed-size* windows to the left and right of proposed change point which use a limited number of samples. In contrast, the BIC criterion uses all the samples on either side of the proposed change point, and is hence robust. Experimental results on 77 seconds of speech from two speakers is shown in Figure 1. Panel (a) plots the first dimension of the cepstral vectors; the dotted line indicates the location of the change. Panels (b) and (c) show the log likelihood ratio and symmetric Kullback-Liebler distances respectively computed with 1 second windows (100 frames) on either side of a hypothesized change point. In both cases a local maximum is attained at the location of the change. However, there are several other maxima which do not correspond to any change points. Panel (d) displays the BIC criterion that clearly predicts the change point.

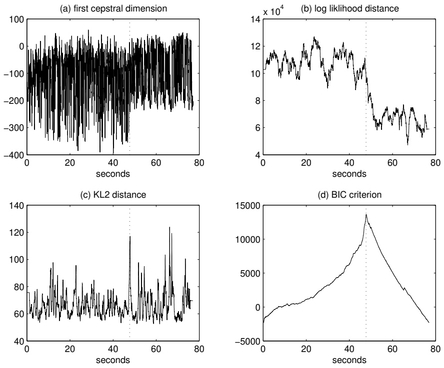

Figure 1: Detecting one changing point

2.2.2 Detecting Multiple Changing Points

The proposed single change point detection scheme can be generalized to detecting multiple change points using the following algorithm.

(1) initialize the interval $[a, b]$: $a = 1; b = 2$.

(2) detect if there is one changing point in $[a, b]$ via BIC.

(3) if (no change in $[a, b]$)
    let $b = b + 1$;
else
    let $\hat{t}$ be the changing point detected;
    set $a = \hat{t} + 1$; $b = a + 1$;
end

(4) go to (2).

By expanding the window $[a, b]$, the final decision is made based on as many samples as possible. Finally note that the BIC scheme while inherently threshold-free can also be viewed as a dynamic thresholding scheme on the log likelihood distance $(\lambda \frac{1}{2}(d + \frac{1}{2}d(d+1)) \log N)$ The accuracy of this procedure depends on the detectability of the true change points. If $\mathcal{T} = \{t_i\}$ are the true change points, the detectability is defined as $D(t_i) = \min(t_i - t_{i-1} + 1, t_{i+1} - t_i + 1)$. Low detectability leads to missed change points which can contaminate the next Gaussian model and thus affect the detection of the next change point. While in principle this algorithm has a quadratic complexity, one can dramatically reduce the cost by performing a crude search with further refinement. An efficient implementation is described in [33].

2.2.3 Change Point Detection on 1997 Hub4 Evaluation Data

Table 1 compares the results of applying this segmentation procedure on the 1997 Hub4 evaluation data (3 hours long) to the hand-segmentation provided by NIST. The scheme detected 462 changes of which 19 (4.1%) were spurious (i.e., occurred within a speaker turn). About 20 (4.3%) detected points had biases as shown in panel (a) in Figure 2. However, these biases are small ($<$ 1 second). The NIST segmentation has 620 changes of which 207 (33.4%) are missed. Of these 154 (25.0%) were short turns ($<$ 2 seconds). Panel (b) shows the histogram of the detectability of the true changes; there were 223 true changes with detectability less than 2 seconds. Panel (c) shows the histogram of

Type-I Error	4.1%		
Type-II Error	33.4%	$\leq 2s$	25.0%
		$> 2s$	8.4%

Table 1: Change detection error rates

	Error Rate
NIST hand-segmentation	19.8%
IBM segmentation	20.4%

Table 2: Segmentation and clustering in Hub4 1997 task

the detectability of the true changes which were missed in the detection; it is clear that most of the errors came from low detectabilities. Panel (d) describes the Type-II error rates according to different degrees of detectability: when detectability is below 1 second, as the type-II error rate is 78%, most such changing points were missed; as the detectability increases, the Type-II error drops. Table 2 compares

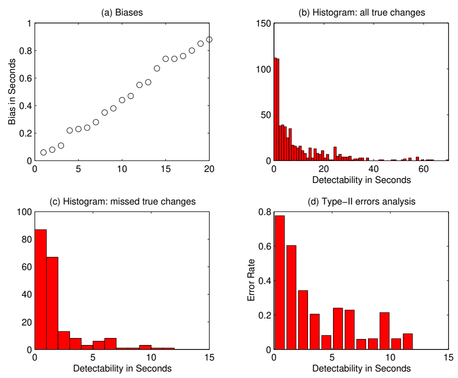

Figure 2: Error analysis of change detection

recognition performance of our segmentation with manual segmentation provided by NIST.

2.3 Clustering via BIC

Clustering can also be viewed as model selection. We have a set of segments with associated data samples. Let $\mathcal{C}_k = \{c_i : i = 1, \cdots, k\}$ be a clustering with k clusters, each c_i having n_i samples. Modeling each c_i as $N(\mu_i, \Sigma_i)$, the BIC value of $\mathcal{C}_k$ is

$$BIC(\mathcal{C}_k) = \sum_{i=1}^{k} \{-\frac{1}{2} n_i \log |\Sigma_i|\} - \lambda P \tag{4}$$

where $P = \frac{1}{2}(d + \frac{1}{2}d(d+1)) \log N$ and $\lambda = 1$. One chooses the clustering with maximal BIC value.

2.3.1 Hierarchical Clustering via greedy BIC

Finding the global best clustering using BIC as above is expensive. However, in hierarchical clustering BIC can be optimized greedily. Consider binary bottom-up clustering with some distance metric. Let $\mathcal{C} = \{c_1, \cdots, c_k\}$ and $\mathcal{C}' = \{c, c_3, \cdots, c_k\}$ be two clusterings, where, in the latter clusters c_1 and c_2 are

	Prepared	Spontaneous
Baseline	18.8%	27.0%
MLLR w/o clustering	18.7%	26.9%
MLLR w/ ideal clustering	17.5%	24.8%
MLLR w/ BIC clustering	17.5%	24.6%

Table 3: MLLR adaptation enhanced by BIC clustering

merged into, say, cluster c. Modeling each c_i as $N(\mu_i, \Sigma_i)$ and c as $N(\mu, \Sigma)$ the increase in BIC value by going from $\mathcal{C}$ to $\mathcal{C}'$ is

$$BIC = n \log |\Sigma| - n_1 \log |\Sigma_1| - n_2 \log |\Sigma_2| - \lambda P \tag{5}$$

where $n = n_1 + n_2$, $P = \frac{1}{2}(d + \frac{1}{2}d(d+1)) \log N$ and $\lambda = 1$. Two nodes are not to be merged if (5) is negative. Since the BIC value is increased at each merge, we are searching for an "optimal" clustering tree by optimizing the BIC criterion in a greedy fashion. Note that BIC is used as a termination criterion. The distance measure in the bottom-up process could be BIC or any other distance measure. Clearly this approach also works for top-down clustering.

2.3.2 Speaker Clustering on the Hub4 1996 evaluation data

We experimented with this algorithm on the 1996 Hub4 evaluation data [20] that had 824 segments from 28 speakers. Bottom-up clustering with log likelihood ratio distance and maximum linkage is used with the BIC termination criterion (5). Our algorithm produces 31 clusters with high purity. Recognition results with un-supervised speaker adaptation using MLLR on these clusters is shown in Table 3. There is negligible degradation in performance relative to ideal speaker clustering.

3 Acoustic Modeling

3.1 Bayesian Information Criterion for Gaussian Mixtures

HMM states in a recognizer are typically modeled using Gaussian mixtures. The number of Gaussians per state is either chosen based on the total number of Gaussians the system can support and/or number of samples in training associated with a particular state. Since too few Gaussians leads to under-fitting and too many to over-fitting, BIC is a natural choice for determining the number of Gaussians for each state. For each state and integer n we have a model with n Gaussians and associated BIC value. One chooses the model with maximal BIC value.

Figure 3 illustrates how this procedure works for a particular HMM state. The horizontal axis represents the number of Gaussians. The vertical axis represents the log-likelihood in Panel (a) and the BIC value in Panel (b). As the number of Gaussians increases, the likelihood always improves, whereas the BIC value first increases and then decreases. The optimal BIC value attained at $n = 27$. The total number of Gaussians in the system (for all the states) is automatically determined with the above procedure. A modified BIC scheme with penalty weight $\lambda \neq 1$ can be used to give models different number of Gaussians for all the states; smaller values of λ lead to larger models.

To compare above procedure to heuristic thresholding a collection of acoustic models of varying (total) sizes ranging from 90K Gaussians to 289K Gaussians were built using different values of λ. A 90K Gaussian model was also built using heuristic thresholding. Recognition results on a subset of the 1997 Hub4 evaluation test set with manual segmentation from NIST is shown in Table 4. For the same total number of Gaussians, the BIC procedure gives a better model. Moreover, BIC gives a sequence of

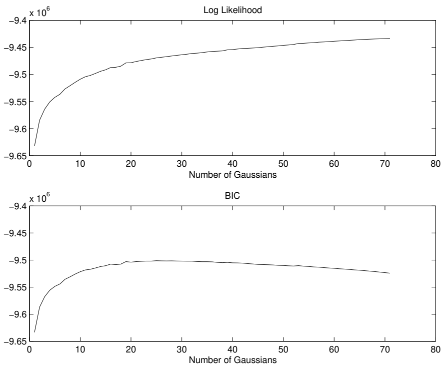

Figure 3: Choosing the number of Gaussians by maximizing the BIC criterion

	# Gaussians	All	F0	F1	F2	F3	F4	F5	FX
Standard	90K	26.0	11.9	23.5	31.7	28.4	28.5	22.3	42.3
$\lambda = 1.00$	90K	25.2	11.6	23.1	30.5	27.7	26.2	20.5	41.8
$\lambda = 0.80$	135K	24.7	11.2	21.2	29.5	29.0	26.8	21.6	41.2
$\lambda = 0.65$	178K	24.2	10.7	21.5	29.3	26.5	25.9	21.4	40.3
$\lambda = 0.54$	237K	23.8	10.7	21.6	29.3	26.5	24.2	19.7	39.6
$\lambda = 0.45$	289K	23.5	10.5	21.5	28.9	24.4	24.6	20.7	39.0

Table 4: Comparison of the BIC approach with the thresholding approach on the 1997 evaluation subset.

better and larger models for decreasing values of λ. In our experiments, compared to thresholding, the BIC procedure tends to favor more Gaussians for complex sounds (vowels) and favor less Gaussians for simple sounds (fricatives).

3.2 Semi-tied Covariance Matrices

Modeling highly correlated features accurately with Gaussian distributions requires full covariances. This requires a large number of parameters and thus restricts the number of Gaussian distributions that can be robustly estimated from a given data-set. This section describes an alternative to full covariance matrices where the covariance matrix is now split into two elements, one full and one diagonal, which may be tied at separate levels. The covariance matrix is of the form

$$\mathbf{\Sigma}^{(m)} = \mathbf{H}^{(r)T}\mathbf{\Sigma}^{(m)}_{\text{diag}}\mathbf{H}^{(r)} \tag{6}$$

where r is the *semi-tied class* of Gaussian component m. Though the use of multiple semi-tied classes has been shown to give some gains on large vocabulary tasks, it was decided to restrict the number of classes to one. This greatly simplifies adaptation of these models (though semi-tied transforms may be adapted as described in [9])

Semi-tied covariances can also be viewed as class-dependent linear feature space transformations and this viewpoint leads to an efficient implementation of semi-tied covariances [9, 32]. In fact, if there is a single semi-tied class, then it reduces to standard diagonal Gaussian models in the transformed space. Thus if the original feature-space (in our case determined by LDA [3]) was $\mathbf{o}$, then the effective space in which models are built is

$$\hat{\mathbf{o}} = \mathbf{H}^{-1}\mathbf{o} = \mathbf{A}\mathbf{o} \tag{7}$$

The task is now to estimate $\mathbf{A}$ in an ML fashion.

3.2.1 Training Semi-Tied Covariance Matrices

A couple of schemes have been used to obtain the semi-tied transform and related model parameters. The one used in this work is to maximize the following expression with respect to $\mathbf{A}$.

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = \sum_{\tau,m} \gamma_m(\tau) \log \left(\frac{|\mathbf{A}|^2}{\left|\mathrm{diag}\left(\mathbf{A}\mathbf{W}^{(m)}\mathbf{A}^T\right)\right|} \right) \tag{8}$$

where

$$\mathbf{W}^{(m)} = \frac{\sum_\tau \gamma_m(\tau) \left(\mathbf{o}(\tau) - \mu^{(m)}\right) \left(\mathbf{o}(\tau) - \mu^{(m)}\right)^T}{\sum_\tau \gamma_s(\tau)} \tag{9}$$

$$\mu^{(m)} = \frac{\sum_\tau \gamma_m(\tau)\mathbf{o}(\tau)}{\sum_\tau \gamma_m(\tau)} \tag{10}$$

For very large numbers of Gaussian components it is not possible to store separate within class covariance matrices for all Gaussian components[1]. For this reason, covariance matrices were stored at the state level, rather than the Gaussian component level. This was found to have very little effect on recognition performance. Having optimized this with respect to $\mathbf{A}$, the mean and the variance of the Gaussian components associated with the HMMs are then found using

$$\hat{\mu}^{(m)} = \frac{\sum_\tau \gamma_s(\tau)\mathbf{A}\mathbf{o}(\tau)}{\sum_\tau \gamma_m(\tau)} \tag{11}$$

and

$$\hat{\mathbf{\Sigma}}^{(m)} = \mathrm{diag}\left(\mathbf{A}\mathbf{W}^{(m)}\mathbf{A}^T\right) \tag{12}$$

After obtaining the new feature-space transformation the HMM parameters are updated using Baum-Welch re-estimation. Table 5 shows experimental results on the planned speech (F0) and spontaneous speech (F1) portions of the 1996 DARPA Hub4 evaluation test. The first experiment used a single semi-tied class, while the second used four classes; one each for the HMM states of the following sounds a) stop-consonants and flaps, b) fricatives, c) vowels and diphthongs, and d) nasals, glides and silence.

3.3 Pronunciation Networks

Typically in a large vocabulary speech recognition system a word is represented as a sequence of its constituent phonemes as defined by a hand-generated lexicon, or as several alternative sequences if multiple pronunciations of the word are allowed.

A major factor in the performance of a speech recognizer is the consistency between the way the speaker pronounces the words and the way that the pronunciation is specified in the dictionary; when there is a mismatch recognition errors are likely to occur.

[1]An alternative optimization does allow training of these very large systems.

Expt	F0 (planned)	F1(spontaneous)
Baseline	21.1	29.1
1 Semi-tied class	19.3	28.4
4 Semi-tied classes	19.4	29.0

Table 5: Semi-tied covariances: % WER on the 1996 Hub4 evaluation data a) Baseline b) Single semi-tied class c) Four semi-tied classes.

One approach to bridging the gap between lexicon and pronounced speech is to alter the dictionary to reflect common pronunciation variations. Alternatively, one may choose to leave the dictionary unchanged and reflect the pronuciation deviations in the acoustic model topology. In this paper we report on an implementation of the latter, an automatic method for discovering an appropriate model for each context-dependent phoneme which allows for such phenomena as reduced pronunciations and substituted phonemes where warranted by observation on training data.

3.3.1 Generating Observations of Strings

For the purposes of discussion, the processing required to build context-dependent pronunciation networks may be divided into two parts: a sequence of "pre-processing" steps resulting in sets of observations from which to train the pronunciation networks and the actual training of those networks. The material described in this section constitutes the preprocessing steps; the discussion in the next section describes the building of the pronunciation networks given the observations.

The observations from which the networks are built are sequences of label strings; each observed label sequence corresponds to the output of a constrained phoneme recognizer for the portion of the waveform aligned to a given phoneme by a Viterbi alignment.

The first step towards the generation of the sets of observation label strings from which each pronunciation network is built, then, is to perform a Viterbi alignment of the text to the speech waveform for each of the training utterances. The canonical pronunciation of each word as defined by the lexicon, with the usual three-state, left-to-right model is used in this computation. The alignment step provides a labeled segmentation of each utterance into phonetic regions, with the labels corresponding to the states in the model for each phoneme.

The second phase of preprocessing is to perform some type of phoneme recognition on the training data. We have chosen to use the same alphabet as in the Viterbi alignment, i.e. thirds-of-phones. Our phoneme recognizer consists of the same acoustic models as the baseline speech recognition system along with a set of transition probabilities among context-dependent phones. Finding the most-likely sequence of phones yields quite clean label sequences. The resulting label sequences are segmented according to the subphonetic boundaries calculated in the Viterbi alignment and pooled according to the state label associated with the alignment.

The third and final phase of preprocessing is to partition the set of observation sequences for each phoneme into context-dependent units. For this we use a decision tree which asks questions about phonetic context and measures the goodness of a split in terms of the reduction in entropy in the distribution over labels. Each leaf in this tree represents a context-dependent unit for which we will build a pronunciation network using the technique outlined in the previous section. Dropping the training data down the tree yields a set of label sequences for each context-dependent unit from which to build the network which will characterize it.

Note that this tree is distinct from that which defines the context-dependent units for which acoustic models are built as will be discussed in the next section.

3.3.2 Pronunciation Models given Observed Strings

The discussion in this section assumes that we have a set of observations in the form of sequences of phonemic labels and that the observations have been partitioned based on phonetic context. These preprocessing steps were described in the previous section. Having obtained the observations, we would like to discover from them the observations a network of states which represents well the collection of observed label sequences for each context; the procedure by which we build a network to represent the observations for each context is the subject of this section.

This procedure is similar in spirit to that described in [31] but differs from their implementation in order to reduce the required computations.

For each context, we first discard all sequences containing a symbol which has not been observed at least N times in the pool of training sequences for this context, where N is a hand-set threshold. We then divide the remaining training observations into two sets, a "development" set for building the initial networks and a second "held-out" set for reducing the number of states in the network.

Next, we build a large initial network which can explain all the sequences in the development training set. We have investigated initializing this network as null and as the default three state left-to-right network and have found the latter to provide slightly better recognition results. Against the starting point of null or the default network, we check the first observation to see if it may be explained by the existing network. If so, we update transition probabilities only. If the observation does not align to the existing network, we add a parallel path consisting of one state each time the phoneme in the sequence differs from its left neighbor, with transition probabilities derived from the number of repetitions of the phonemes within the observation. For each remaining observation in the development training pool we repeat the procedure, checking whether it may be explained by the existing network; if so we update transition probabilities and if not we add a parallel path to the network. After all observations have been incorporated into the network, we prune all branches with transition probability into the branch less than ϵ where ϵ is a hand-determined threshold.

Having built the initial network, we begin collapsing states where such merges are favorable on our held-out training data. Each state in the network is labeled from the alphabet of subphonemic units, e.g. AA_1. Only states which carry the same label are considered for merge. The probability distribution over labels for each state is concentrated entirely on the associated label; the likelihood of the held-out data is computed before and after a given merge. If the likelihood increases, or if the number of observations which can be modeled by the resulting network is larger than was the case prior to the merge, the merge is retained; otherwise the two states are kept distinct. Because states are collapsed whenever an advantageous merge is found, the order in which states are evaluated for merging impacts the final network. We have somewhat arbitrarily started with the first two states having the same label and hold the first state fixed, sequentially evaluating the second state until a good merge is found. Once a merge is retained, we reset the starting state to be the next state in the network from the current starting state and iterate until no more good merges exist or until there are fewer than S states in the network, where S is a threshold set by hand.

Merging of two states consists of creating a new state whose parents are the union of the parents of the two states and whose children are itself plus the children of the two states excluding those states themselves (self-loops) and deleting the two unmerged states and their incoming and outgoing arcs.

Finally, two iterations of the E.M. algorithm are run to estimate transition probabilities for the network.

3.3.3 Using Pronunciation Networks in Recognition

The error rates resulting from using the pronunciation networks to define the context-dependent phoneme model topologies are compared with the errors resulting from a baseline three-state, left-to-right model topology which has no skips. We report the error rates on both the Wall Street Journal and Broadcast News databases. On WSJ, the baseline of 8.0% error fell to 7.4% when the pronunciation network topologies were used in lieu of the default networks. On the Broadcast News data the error rate was again reduced by using the pronunciation networks, as detailed in table 6. Appealingly, the largest reduction in error occurs in the case of spontaneous speech, a condition where we would expect that pronunciations variations from the lexical representation might be larger than what would be observed in the case of planned speech, and therefore a condition for which this type of modeling would be potentially valuable.

	F0	F1	F2	F3	F4	F5	FX
A	9.1	20.8	28.0	25.1	24.4	19.6	37.1
B	8.9	20.1	27.8	25.0	24.4	19.5	37.4

Table 6: Error rates on Broadcast News data. Condition A is the baseline. Condition B uses pronunciation networks. F0=Clean/Planned Speech. F1=Clean Spontaneous Speech. F2=Speech on Telephone Channels. F3=Speech with Background Music. F4=Speech in Degraded Acoustics. F5=Non-native speakers. FX=Combinations of F1-F5.

3.4 Tail Distribution Modeling

An examinination of the histograms of any single dimension of the acoustic data assigned to a particular HMM state shows that the distributions are a) skew–symmetric, b) peakier than typical Gaussians and c) have tails that taper off at a slower rate than for a Gaussian tail. Despite these limitations, mixtures of Gaussians perform well in speech recognition experiments. Addressing these issues may lead to improved performance. An option (that increses the number of parameters dramatrically is) to increses the number of Gaussian components in the mixtures. This section describes two efficient alternatives that address b) and c) - Richter distributions and power-exponential distributions. A Richter distribution ([19, 6]) is a mixture of Gaussians with the same mean and covariances that are multiples of each other. A power exponential distribution is obtained by raising the exponent in a Gaussian distribution to a power that is possibly different from that of the Gaussian. For large powers they are similar to a uniform distribution whereas for small powers they have have sharp peaks and heavy tails [5].

3.4.1 Richter Distributions

Richter distributions are described by

$$f(\mathbf{o}; \mu, \mathbf{\Sigma}, p(v)) = \int \mathcal{N}(\mathbf{o}; \mu, v^2\mathbf{\Sigma}) p(v) dv \ , \tag{13}$$

where $p(v)$ is a probability density function and includes Gaussian and Cauchy distributions as special cases. For example, $p(v) = \delta_1(v)$ (where $\delta_{v_r}(v)$ is the Kronecker delta function) corresponds to a standard Gaussian distribution. By appropriately choosing $p(v)$ the tails and the peakiness of the distribution can be controlled. An EM scheme is described for ML estimates of μ and $\mathbf{\Sigma}$ that does not require explicitly obtaining the distribution $p(v)$ is described in [12]. A discrete version of (13) is described in [19] where $p(v) = \sum_r w_r \delta_{v_r}(v)$ with $w_r > 0$, $\sum_r w_r = 1$. The paper also gives formulae for

ML estimates of μ, $\mathbf{\Sigma}$ and w_r. This form of distribution was used in [6] for discrete speech modeling, though in the experiments described the discrete distribution of v was determined a priori rather than trained from the data.

This paper proposes using Richter mixtures for modeling HMM states in LVCSR systems.

$$\mathcal{L}(\mathbf{o}) = \sum_m \sum_r w_r^{(m)} \mathcal{N}(\mathbf{o}; \mu^{(m)}, v_r^{(m)2} \mathbf{\Sigma}^{(m)}) \ . \tag{14}$$

It is possible to tie the Richter distribution parameters $\mathbf{w}^{(m)}$ and $\mathbf{v}^{(m)}$ over many Richter distributions. In our experiments globally tied Richter distribution parameters were obtained using the Hub4 training data. The tails of the Richter distribution are longer than those of the Gaussian distribution, indicating that, in a likelihood sense, the Gaussian components are sub–optimal.
The following re–estimation formulae (modified versions of those in [6]) are used

$$\hat{\mu}^{(m)} = \frac{\sum_{r,\tau} \frac{\gamma_r^{(m)}(\tau)}{\hat{v}_r^{(pm)2}} \mathbf{o}(\tau)}{\sum_{r,\tau} \frac{\gamma_r^{(m)}(\tau)}{\hat{v}_r^{(pm)2}}} \ , \qquad \hat{\mathbf{\Sigma}}^{(m)} = \frac{\sum_{r,\tau} \frac{\gamma_r^{(m)}(\tau)}{\hat{v}_r^{(pm)2}} \hat{\mathbf{W}}^{(m)}(\tau)}{\sum_{r,\tau} \gamma_r^{(m)}(\tau)} \ , \tag{15}$$

$$\hat{v}_r^{(p)2} = \frac{\sum_{M^{(p)},\tau} \gamma_r^{(m)}(\tau) \hat{q}^{(m)}(\tau)}{d \sum_{M^{(p)},\tau} \gamma_r^{(m)}(\tau)} \ , \quad \text{and} \quad \hat{w}_r^{(m)} = \hat{c}^{(m)} \frac{\sum_\tau \gamma_r^{(m)}(\tau)}{\sum_{M^{(p)},r,\tau} \gamma_r^{(m)}(\tau)} \tag{16}$$

where

$$\hat{c}^{(m)} = \frac{\sum_{r,\tau} \gamma_r^{(m)}(\tau)}{\sum_{S^{(m)},r,\tau} \gamma_r^{(m)}(\tau)} \ , \qquad \hat{q}^{(m)}(\tau) = \left(\mathbf{o}(\tau) - \hat{\mu}^{(m)}\right)^T \hat{\mathbf{\Sigma}}^{(m)-1} \left(\mathbf{o}(\tau) - \hat{\mu}^{(m)}\right)$$

and

$$\hat{\mathbf{W}}^{(m)}(\tau) = (\mathbf{o}(\tau) - \hat{\mu}^{(m)})(\mathbf{o}(\tau) - \hat{\mu}^{(m)})^T \ . \tag{17}$$

$M^{(p)}$ is the set of components sharing the same Richter parameters, p_m is the Richter class of component m, d is the dimensionality of the observation vector $\mathbf{o}(\tau)$ and $\gamma_r^{(m)}(\tau)$ is the posterior probability of being in Richter component r of component m at time τ and $S^{(m)}$ is the set of components in the same state as m. Formulae (15)–(16) yield an iterative estimation scheme since the mean and the variance are a function of $\hat{\mathbf{v}}^{(r)}$, which itself is a function of the estimates of the mean and variance. The sufficient statistics for this operation are the occupancy, sum and sum squared of the feature vector for each Richter distribution of each component. Thus if there are M components and R Richter distributions per component, the equivalent of $M \times R$ components must be stored. An alternative to this and the one used in this section is to either update the Richter distribution parameters or the means and variances. In this case it is only necessary to store parameters at the Richter tying level or the component level.
One reason for using Richter distributions rather than additional Gaussian components is the efficiency of the likelihood calculation. Indeed,

$$\mathcal{L}(\mathbf{o}(\tau); m) = \sum_{m,r} b_r^{(m)} \exp\left(-\frac{q^{(m)}(\tau)}{2 v_r^{(m)2}}\right) \ ,$$

where $q^{(m)}(\tau)$ is a function of the component, m, and observation

$$q^{(m)}(\tau) = \left(\mathbf{o}(\tau) - \mu^{(m)}\right)^T \mathbf{\Sigma}^{(m)-1} \left(\mathbf{o}(\tau) - \mu^{(m)}\right) \tag{18}$$

and $b_r^{(m)}$ is a function of the Richter component r, but independent of the observation

$$b_r^{(m)} = \frac{1}{\sqrt{2^d\pi^d|\det\mathbf{\Sigma}^{(m)}|}}\frac{w_r^{(m)}}{\sqrt{v_r^{(m)2d}}} \; ,$$

The additional cost is in the log–add over the Richter components. This may be ignored if a max of the components is taken, rather than the sum.
It is also common to use linear transformations to adapt model parameters to be more representative of a particular speaker, or acoustic environment. A variety of linear transformations and re–estimation formulae are described in [8]. Modifying these formulae to handle Richter distributions is trivial. The main modification is to deal with $\frac{\gamma_r^{(m)}(\tau)}{v_r^{(m)2}}$ rather than the standard posterior component probability. As an example the estimation formulae for the transform $\hat{\mathbf{A}}$ in maximum likelihood linear regression, where Richter components are used, is considered. Here

$$\hat{\mu}^{(m)} = \mathbf{A}\mu^{(m)} + \mathbf{b} = \mathbf{W}\xi^{(m)} \tag{19}$$

where $\xi^{(m)}$ is the extended mean vector $\left[\begin{array}{c} \mu^{(m)T} 1 \end{array}\right]^T$. The estimation formula for row i of the transformation matrix is

$$\hat{\mathbf{w}}_i = \mathbf{k}^{(i)}\mathbf{G}^{(i)-1} \; , \tag{20}$$

where

$$\mathbf{G}^{(r)} = \sum_{M,\tau,r} \frac{\gamma_r^{(m)}(\tau)}{v_r^{(m)2}\sigma_i^{(m)2}}\xi^{(m)}\xi^{(m)T} \quad \text{and} \quad \mathbf{k}^{(i)} = \sum_{M,\tau,r} \frac{\gamma_r^{(m)}(\tau)}{v_r^{(m)2}\sigma_i^{(m)2}}\xi^{(m)T}o_i(\tau) \; . \tag{21}$$

Similarly modifications to the variance adaptation formulae are possible.

3.4.2 Power Exponentials

Consider the class of densities

$$f(\mathbf{o};\mu,\mathbf{\Sigma},\alpha) = \rho_\alpha|\det\mathbf{\Sigma}|^{-1/2}\exp(-(\gamma_\alpha q)^{\alpha/2}), \tag{22}$$

where

$$q = (\mathbf{o}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{o}-\mu) \; , \tag{23}$$

$$\gamma_\alpha = \frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)} \quad \text{and} \quad \rho_\alpha = \frac{\alpha\Gamma^{\frac{1}{2}}(3/\alpha)}{2\Gamma^{\frac{3}{2}}(1/\alpha)} \; . \tag{24}$$

This class was recently suggested and studied in [5]. The one dimensional case appears to have first been suggested by Subbotin, [14]. The class (22) will be referred to as the the power exponential distribution. It is also known as the error function, p–Gaussians or as α–Gaussians.

Following (14) a model is considered where each state in the system is modelled by a mixture of power exponential distribution, i.e.

$$\mathcal{L}(\mathbf{o}) = \sum_m w^{(m)} f(\mathbf{o};\mu^{(m)},\mathbf{\Sigma}^{(m)},\alpha^{(m)}). \tag{25}$$

It is worth noticing that the class of functions described in (22) is not a subset of the class described in (13). Power exponential distributions can not in general be modelled with Richter distributions. This fact can be verified by noticing that functions in the class (13) are all log concave, whereas the power exponentials are not log concave for $0 < \alpha < 1$. This makes the framework of [12] unsuitable for parameter update for $0 < \alpha < 1$.

The estimation formula for $w^{(m)}$ is identical to the standard HMM re–estimation formulae. Update formulae for $\mu^{(m)}$ and $\mathbf{\Sigma}^{(m)}$ are suggested in [5]:

$$\hat{\mu}^{(m)} = \frac{\sum_\tau \gamma^{(m)}(\tau) \left(q^{(m)}(\tau)\right)^{\alpha^{(m)}/2-1} \mathbf{o}(\tau)}{\sum_\tau \gamma^{(m)}(\tau) \left(q^{(m)}(\tau)\right)^{\alpha^{(m)}/2-1}}, \tag{26}$$

and

$$\hat{\mathbf{\Sigma}}^{(m)} = \frac{\sum_\tau \gamma^{(m)}(\tau) \left(q^{(m)}(\tau)\right)^{\alpha^{(m)}/2-1} \hat{\mathbf{W}}^{(m)}(\tau)}{\sum_\tau \gamma^{(m)}(\tau)}, \tag{27}$$

where $q^{(m)}(\tau)$ is defined in equation (18), $\hat{\mathbf{W}}^{(m)}(\tau)$ in equation (17) and $\gamma^{(m)}(\tau)$ is defined to be the posterior probability of being in the power exponential component m at time τ. It is not known that the overall likelihood is guaranteed to increase with the update given by (26)–(27), but numerical evidence suggests that this is true.

Special consideration for $0 < \alpha < 1$ is suggested in [5]. The powers $\alpha^{(m)}$ can either be fixed on a global level or they can be updated according to the formula given in [5]:

$$\hat{\alpha}^{(m)} = \text{argmax}_\alpha \sum_\tau \gamma^{(m)}(\tau) \log \left(f(\mathbf{o}(\tau); \hat{\mu}^{(m)}, \hat{\mathbf{\Sigma}}^{(m)}, \alpha)\right) \tag{28}$$

With this update of $\alpha^{(m)}$ the likelihood is guaranteed to increase. Figure 4 shows the distribution

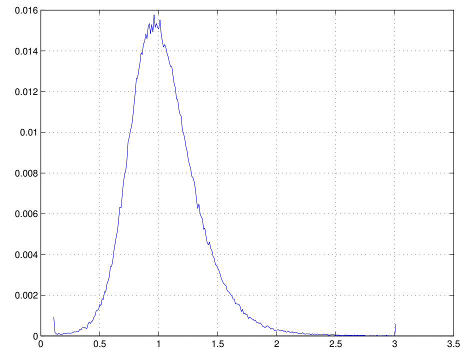

Figure 4: The distribution of powers, α, after training using (28) on the Hub4 1997 data.

of α estimated on a per component case. The mean of the values of α is approximately one. It is interesting to note that the Gaussian component equivalent of power exponential components, $\alpha = 2$, occurs infrequently. Again, this indicates that Gaussian components are sub–optimal in a likelihood sense.

Currently adaptation of power exponentials have not been investigated.

3.4.3 Tail-distribution Modeling - Results

The two forms of modified tail distribution modeling were investigated on the 1998 Hub4 partitioned evaluation test set. The baseline system for the Richter components had a total of about 135,000 components. A 4 distribution Richter component system (R=4) was initialised using the means and variances of the baseline system. The Richter parameters were tied at the state level. Table 7 shows the

Richter System	Error Rate (%)		
	F0	F1.	Avg
base	11.6	18.5	18.7
base+adapt	10.1	17.0	16.4
Richter	11.3	18.1	18.4
Richter+adapt	10.1	16.9	16.3

Power Exponential System	Error Rate (%)		
	F0	F1.	Avg
base ($\alpha = 2$)	11.8	22.9	26.1
$\alpha = 1$	11.5	23.0	25.5
EM update for α	11.9	22.6	25.4

Table 7: Results on the Hub4 1997 partitioned evaluation test set

comparison of a Richter system and the equivalent baseline system. The adaptation scheme used in both was a global mean and full variance transform described in [8]. This was applied in an unsupervised batch adaptation mode. Using Richter components showed a small gain in performance over the standard Gaussian components. After adaptation the performance of the two systems are indistinguishable.

The experiments using power exponential components used a modified baseline system consisting of approximately 120,000 Gaussians. The test was performed on a subset of the 1997 partitioned evaluation that was used for development [22]. Finally a smaller language model than for that of the tests with the Richter distribution where used, thus degrading the performance for the spontaneous speech category, F1, and for some of the more difficult conditions, F2–FX. Two power exponential systems were built. The first used a fixed value of $\alpha^{(m)} = 1$ for all components, motivated by figure 4. The second system used a per–component value of $\alpha^{(m)}$ obtained using equation (28).

Table 7 shows the performance of the various power exponential systems. Again only small reductions in word error rate were observed using the improved tail modeling.

4 Adaptive Training

Acoustic models trained on non-homogeneous data have large variance since they model both *inter-speaker* and *intra-speaker* variability. Assuming some form of adaptation is used during testing, it is preferable to adapt a canonical model that just represents the intra-speaker variability. This paper addresses two forms of adaptive training. The first is a modified form of the so-called SAT (speaker adaptive training) scheme [34], where a constrained-model space transform is used instead of MLLR for adaptation. This yields simple estimation formulae for the canonical model. The second form of adaptive training uses a simple form of transform for adaptation with a more complex canonical model.

As with all adaptive training schemes the training procedure is as follows:

1. Partition the training data into “appropriate” groups.
2. Estimate the adaptation transform for each partitioned subset of the training data.
3. Estimate the canonical model given the partition adaptation transforms.
4. Goto (2) unless convergence criteria are satisfied.

Each of the estimation stages is described for both forms of adaptive training considered.

4.1 Modified Speaker Adaptive Training

Modified speaker adaptive training (MSAT) uses a constrained model-space transform. Here

$$\hat{\mu}^{(sm)} = \mathbf{A}'^{(r)}\mu_{\mathbf{a}}^{(m)} + \mathbf{b}'^{(r)} \tag{29}$$

$$\hat{\Sigma}^{(m)} = \mathbf{A}'^{(r)}\mathbf{\Sigma}^{(m)}\mathbf{A}'^{(r)T} \tag{30}$$

Since the majority of speech recognition systems, and the ones considered in this paper, use diagonal covariance matrices, this form of transform may be efficiently implemented as multiple transformations of the feature-space. Thus

$$\mathcal{L}(\mathbf{o};\mu_{\mathbf{a}}^{(m)},\mathbf{\Sigma}^{(m)},\mathbf{A}'^{(r)}) = \frac{1}{|\mathbf{A}^{(r)}|^2}\mathcal{N}(\mathbf{A}^{(r)}\mathbf{o} + \mathbf{b}^{(r)}\mu_{\mathbf{a}}^{(m)},\mathbf{\Sigma}^{(m)}) \tag{31}$$

where $\mathbf{A}^{(r)-1} = \mathbf{A}'^{(r)}$ and $\mathbf{b}^{(r)} = -\mathbf{b}^{(r)}$.

4.1.1 Adaptation Transform

The estimation of the adaptation transform for a particular partition of the training data[2] is obtained in a simple iterative fashion.

$$\mathbf{w}_i = \left(\alpha\mathbf{p}_i + \mathbf{k}^{(i)}\right)\mathbf{G}^{(i)-1} \tag{32}$$

where $\mathbf{p}_i$ is the extended cofactor row vector $\begin{bmatrix} 0 & c_{i1} & \dots & c_{in} \end{bmatrix}$, $(c_{ij} = \mathrm{cof}(\mathbf{A}_{ij}))$,

$$\mathbf{G}^{(i)} = \sum_m \frac{1}{\sigma_i^{(m)2}}\sum_\tau \gamma_m(\tau)\zeta(\tau)\zeta(\tau)^T \tag{33}$$

$$\mathbf{k}^{(i)} = \sum_m \frac{1}{\sigma_i^{(m)2}}\mu_{\mathbf{a}i}^{(m)}\sum_\tau \gamma_m(\tau)\zeta(\tau)^T \tag{34}$$

and α satisfies a simple quadratic expression given in [8].

4.1.2 Canonical Model

In [8] the estimation of the canonical model is given by

$$\hat{\mu}_{\mathbf{a}}^{(m)} = \frac{\sum_{s,\tau}\gamma_m^{(s)}(\tau)\hat{\mathbf{o}}^{(rs)}(\tau)}{\sum_{s,\tau}\gamma_m^{(s)}(\tau)} \tag{35}$$

and

$$\hat{\mathbf{\Sigma}}^{(m)} = \frac{\sum_{s,\tau}\gamma_m^{(s)}(\tau)(\hat{\mathbf{o}}^{(rs)}(\tau)-\hat{\mu}^{(m)})(\hat{\mathbf{o}}^{(rs)}(\tau)-\hat{\mu}^{(m)})^T}{\sum_{s,\tau}\gamma_m^{(s)}(\tau)} \tag{36}$$

where

$$\hat{\mathbf{o}}^{(rs)}(\tau) = \mathbf{A}^{(rs)}\mathbf{o}(\tau) + \mathbf{b}^{(rs)} \tag{37}$$

and $m \in M^{(r)}$. Thus the sufficient statistics required to estimate the canonical model have the same complexity as the standard Gaussian component estimation schemes.

[2]During testing exactly the same procedure is used for the particular partition of the test data. Of course during testing the canonical model is not updated.

4.2 Cluster Adaptive Training

In cluster adaptive training (CAT) the set of Gaussian component means associated with a particular partition of the training data is given by a linear interpolation of a set of P cluster means [10]. Thus CAT is defined as follows. For a particular Gaussian component, $m \in M_{\mathbf{w}}^{(r_m)}$, the mean for speaker s is given by

$$\mu^{(sm)} = \mathbf{M}^{(m)}\lambda^{(sr_m)} \tag{38}$$

where $\mathbf{M}^{(m)}$ is the matrix of P cluster mean vectors for component m

$$\mathbf{M}^{(m)} = \left[\begin{array}{ccc} \mu_{\mathbf{c}}^{(m1)} & \ldots & \mu_{\mathbf{c}}^{(mP)} \end{array}\right] \tag{39}$$

where $\mu_{\mathbf{c}}^{(mp)}$ is the mean of Gaussian component m associated with cluster p and the cluster weight vector, $\lambda^{(sr_m)}$, for speaker s is given by (assuming that a bias cluster is used)

$$\lambda^{(sr_m)} = \left[\begin{array}{cccc} \lambda_1^{(sr_m)} & \ldots & \lambda_{P-1}^{(sr_m)} & 1 \end{array}\right]^T \tag{40}$$

and r_m is the cluster weight class of Gaussian component m.

4.2.1 Adaptation Transform

$$\lambda^{(sr)} = \mathbf{G}_{\mathbf{w}}^{(sr)-1}\mathbf{k}_{\mathbf{w}}^{(sr)} \tag{41}$$

where

$$\mathbf{G}_{\mathbf{w}}^{(sr)} = \sum_{m \in M_{\mathbf{w}}^{(r)},\tau} \gamma_m(\tau)\mathbf{M}^{(m)T}\mathbf{\Sigma}^{(m)-1}\mathbf{M}^{(m)} \tag{42}$$

$$\mathbf{k}_{\mathbf{w}}^{(sr)} = \sum_{m \in M_{\mathbf{w}}^{(r)}} \mathbf{M}^{(m)T}\mathbf{\Sigma}^{(m)-1}\sum_{\tau}\gamma_m(\tau)\mathbf{o}(\tau) \tag{43}$$

4.2.2 Canonical Model

The canonical model associated with CAT has a different form to the canonical models used in SAT or MSAT. Here

$$\mathcal{M} = \left\{\{\mathbf{M}^{(1)},\ldots,\mathbf{M}^{(M)}\},\{\mathbf{\Sigma}^{(1)},\ldots,\mathbf{\Sigma}^{(M)}\}\right\} \tag{44}$$

These parameters are estimated using

$$\mathbf{M}^{(m)T} = \mathbf{G}^{(m)-1}\mathbf{K}^{(m)} \tag{45}$$

$$\mathbf{\Sigma}^{(m)} = \operatorname{diag}\left(\frac{\mathbf{L}^{(m)} - \mathbf{M}^{(m)}\mathbf{K}^{(m)}}{\sum_{s,\tau}\gamma_m(\tau)}\right) \tag{46}$$

where

$$\mathbf{G}^{(m)} = \sum_{s,\tau}\gamma_m(\tau)\lambda^{(sr_m)}\lambda^{(sr_m)T} \tag{47}$$

$$\mathbf{K}^{(m)} = \sum_{s,\tau}\gamma_m(\tau)\lambda^{(sr_m)}\mathbf{o}(\tau)^T \tag{48}$$

$$\mathbf{L}^{(m)} = \sum_{s,\tau}\gamma_m(\tau)\mathbf{o}(\tau)\mathbf{o}(\tau)^T \tag{49}$$

4.3 Adaptive Training - Results

The standard baseline system for the adaptive training experiments were used. In order to get a valid set of comparisons a set of initial experiments were carried out using this initial model set and the standard linear adaptation schemes. Table 8 shows the results on the Hub4 1997 partitioned evaluation test set.

System	Error Rate (%)		
	F0	F1.	Avg
base	11.6	18.5	18.7
base+mllr	10.7	18.0	17.2
base+constr	10.6	17.0	17.2
base+mllr+cov	10.1	17.0	16.4

Table 8: Results for adapting the standard SI system on the Hub4 1997 partitioned evaluation test set

The baseline recognition rate was 18.7% and this was used to give the adaptation hypothesis and initial alignments to obtain the transforms. Using a single MLLR transform, which has 3660 parameters, the error rate was reduced by about 8%. Similar performance was obtained using the constrained model-space transform. Using MLLR in conjunction with a full variance transform an additional 5% reduction in word error rate was achieved. It is worth noting that this transform is a generalization of both MLLR and the constrained model-space transform.

For the adaptive training experiments the partitioning of the training data was as follows. For the training data labeled with both speaker and focus conditions the partitions were on in terms of these speaker/focus pairings. Where the focus condition was not given, data was simply grouped according to the speaker.

For the MSAT experiments a single full transformation matrix with bias was estimated for all partitions with greater than 1000 frames. For those partitions having less than a 1000 frames a diagonal transformation matrix with bias was estimated. The motivation for this is that the the single global semi-tied transform will roughly de-correlate the data for all components. During recognition a single full transform with bias was estimated for each test partition. Table 9 shows the recognition performance

System	Error Rate (%)		
	F0	F1.	Avg
msat+constr	10.1	16.3	16.4
msat+mllr+cov	9.9	16.2	15.9

Table 9: Results for the MSAT system on the Hub4 1997 partitioned evaluation test set

of adaptation of MSAT system. Using a single full constrained model-space transform a 12% reduction in word error rate was achieved compared to baseline and a 5% reduction compared to using the same adaptation scheme on the baseline SI model. Rather than using a constrained model-space transform the MLLR plus full variance transform could also be used. This gave an overall 15% reduction over the baseline system. However the gain from adapting an adaptively trained system over adapting the baseline SI system was only 3%. This is expected, since as the complexity, and power, of the transform increases the gain from using adaptive training is reduced.

For the CAT experiments four initial clusters were generated. These were based on the data with speaker and focus condition information and were grouped as *Male/Female* with *Clean/Noisy*. MLLR was used to transform the baseline SI model set to each of the groupings. A single interpolation weight was then computed for each training set partition using the alignments from the baseline SI model set. The canonical model parameters were then updated. During recognition a single set of interpolation weight was calculated for each test partition. Table 10 shows the performance of the CAT system

System	Error Rate (%)		
	F0	F1.	Avg
cat	11.0	17.3	17.6
cat+mllr+cov	10.0	16.1	15.7

Table 10: Results for CAT system on the Hub4 1997 partitioned evaluation test set

on the Hub4 1997 partitioned evaluation test set. Using only CAT, requiring estimation of only four parameters, a 6% reduction in word error rate was achieved. Though quite small it is only about 2% worse than using MLLR whilst requiring a factor of about 1000 fewer parameters to be estimated. Rather than using CAT alone it is possible to combine it with other more powerful adaptation schemes. Using MLLR plus a full covariance transform in addition to CAT yielded a 16% reduction in word error rate over the baseline and a 4% reduction over using the same transform on the baseline SI model.

5 Language Modeling

The LM training corpus for Broadcast News provided by LDC has billions of words spanning several years. Cleaning up this data for building LMs requires filters that depend on the source and period the data spans. Moreover, processing time of LMs may depend non-linearly on the length of the corpus (e.g., LMs with classes based on MMI criteria [7]). Our approach therefore was to partition the training corpus into groups, build several LMs for each group, and integrate these LMs using a mixture model. The corpus was partitioned into four groups: text from 1996, 1997 and 1998 and the text of the acoustic training data. Three types of LMs - 3-gram, 4-gram language models with deleted interpolation and a maximum entropy class based language model ([11]) - were built on each partition. The mixture weights were obtained based on either word-error-rate or perplexity minimization on a development test set. Clearly the word-error-rate minimization technique is more expensive and was done in a greedy fashion by starting with two components and gradually increasing the number of LM components; however, both techniques lead to LMs with similar performance on test data. The results with several mixtures sizes based on these two types of optimization (WER and perplexity) are shown in Table 11, where, for comparison, results with the single best LM component - a 4-gram LM built on the 1996 training text is also shown.

6 Some Additional Techniques

6.1 Rover

Recently a scheme that combines hypothesis transcripts from several recognizers was shown to greatly improve the accuracy [18]. The scheme - *recognition output voting error reduction* or ROVER - aligns a set of hypothesis transcripts using dynamic programming and picks the most frequent word (weighted

Lang. Model	Set1	Set2
6-LM tun	18.8	15.7
8-LM tun	18.0	14.9
8-LM perp	18.0	14.9
14-LM perp	17.8	14.7
4g-LM 96	20.2	17.3

Table 11: Mixture LM and single LM

with confidence scores if available) among words aligned to each other; ties are resolved arbitrarily. If the hypothesis transcripts are ordered (e.g., the "best" transcript first), the induced order can be used to resolve the tie among words.

An interesting fact - that deletions are more probable that substitution by *a* particular word - can be used to improve ROVER. To see this, consider an enumeration of possible alignments from a best-first ordered set of three hypothesis transcripts: (w_1, w_1, w_1), (w_1, w_1, w_2), (w_1, w_2, w_1), (w_2, w_1, w_1), $(w_1, w_1, \emptyset)$, $(w_1, \emptyset, w_1)$, $(\emptyset, w_1, w_1)$, $(w_1, \emptyset, \emptyset)$, $(\emptyset, w_1, \emptyset)$, $(\emptyset, \emptyset, w_1)$, $(w_1, w_2, \emptyset)$, $(w_1, \emptyset, w_2)$, $(\emptyset, w_2, w_1)$ and (w_1, w_2, w_3). In all but three of the cases viz., $(w_1, w_2, \emptyset)$, $(w_1, \emptyset, w_2)$ and $(\emptyset, w_2, w_1)$, ROVER picks the best alternative. However, in these three cases, a deletion is to be preferred by the fact above. One way to accomplish this in ROVER is to add a fourth $\emptyset$ script in the alignment process. Depending on the number of hypothesis transcripts one or more $\emptyset$ hypothesis can be added to improve performance. Table 12 shows the performance of several systems used by IBM in the 1998 Hub4 evaluation along with ROVER numbers on subsets of these systems using this idea.

System #	Short description	WER I	WER II
1	289K baseline	15.7%	13.3%
2	Smaller set of phonemes	16.3%	14.7%
3	Telephone models	18.4%	16.6%
4	Left context models	17.5%	14.9%
5	Power Exponential Densities	18.9%	16.0%
6	Speaker Adapted Training	15.5%	12.8%
7	Cluster Adapted Training	15.4%	13.1%
	rover$(7, 6, 1, 2, 3, 4, 5, \emptyset, \emptyset)$	14.5%	12.4%
	rover$(7, 2, 3, \emptyset)$	14.3%	12.5%
	rover$(7, 2, 3)$	14.8%	12.7%
	rover$(7, 6, 2, 3, \emptyset, \emptyset)$	14.4%	12.4%

Table 12: Enumeration of individual systems used in the 1998 hub4 evaluation and some corresponding rover combinations. WER I and WER II refers to the error rates of the two individual test sets in the 1998 hub4 evaluation.

Decoding	All	F0	F1
M_0+Baseline	21.2	13.0	34.8
M_0+Modified	19.9	12.0	31.1
M_0+MLLR	20.5	12.5	33.9
M_0+MLLR+Modified	19.1	11.4	31.1
M_2+Baseline	23.3	13.0	35.0
M_2+MD	21.5	12.0	31.2
M_2+MLLR+Modified	20.9	10.7	30.4
Set1+Baseline	18.0	8.9	19.9
Set1+M_2+Modified	17.7	8.7	19.1

Table 13: Decoding results for M_0 and M_2 speech metrics

6.2 Fast Speaking Rate

Fast speech is well-known to significantly impair recognition accuracy [13]. This paper proposes two approaches - phone models with skip arcs, and faster-rate signal processing to handle fast speech. In the former, standard 3-state left-to-right phone models are modified allowing a skip from the first to the third state. In the latter, the frame rate is increased to 110 from 100 per second. On test data segments are labeled as fast or normal, with the fast segments transcribed using modified phones at the fast rate.

Speaking rate is determined using the average duration of phones. On the training data using Viterbi alignment this information is obtained for both fast and normal speakers. A subset of phones where there is marked difference in duration was identified (say $\mathcal{P}$) and this subset included almost all the consonants and some vowels. During testing each segment of speech is aligned to an initial pass transcription. Each phone in $\mathcal{P}$ is labeled fast or normal by comparing its duration to that in training. Let N^n and N^f denote the number of phones in $\mathcal{P}$ labeled normal and fast respectively. Then, for an integer s we define the metric M_s such that an entire segment is labeled fast or normal based on $N^n - N^f < s$. The metric M_s, given the difficulty of estimating speaking rate (e.g. [15], [16]), is motivated solely by the fact that that it led to good performance improvements on speech recognition using modified phone models and faster signal processing. Table 13 shows experimental results using this approach with and without un-supervised adaptation (MLLR) for two metrics viz., M_0 and M_2 on Set1 in the 1998 Hub4 evaluation test. Rows in the Table with M_0 correspond to word error rates on that subset of segments of Set1 that were labeled as fast; similarly for M_2. The last row shows that on the entire Set1 test set about 0.3% absolute improvement is obtained using this technique.

7 Summary

Transcription of broadcast news is an extremely challenging problem. This paper has described several techniques that are useful in building a state-of-the art speech recognition system for broadcast news. Specifically the paper proposes BIC for segmentation, clustering and choosing the number of Gaussians in a mixture model, pronunciation modeling for conversational speech, semi-tied covariance modeling, tail-distribution modeling, adaptive training (modified SAT and Cluster Adaptive Training) to handle the heterogeneity in Broadcast News data.

8 Acknowledgements

The authors would like to thank M. Monkowski and M. Franz for many valuable discussions on fast speech problems and language modeling.

References

[1] X. Aubert et al., "Improved hidden Markov models for speech recognition," Philips Journal of Research 43, pp 254–245, 1988.

[2] L. R. Bahl et al., "Robust Methods for using Context-Dependent features and models in a continuous speech recognizer", Proc. ICASSP, 1994.

[3] L. R. Bahl et al., "Performance of the IBM large vocabulary continuous speech recognition system on the ARPA Wall Street Journal task", Proc. ICASSP, pp 41-44, 1995.

[4] R. Bakis et al., Transcription of BN Shows with the IBM LVCSR System", Proc. DARPA Sp. Reco. Workshop, 1997.

[5] S. Basu et al., "Power exponential densities for the training and classification of acoustic feature vectors in speech recognition," *Research report*, T.J.Watson Research Center, 1999.

[6] P. Brown, "The Acoustic-Modeling Problem in Automatic Speech Recognition," Ph.D. Thesis, IBM T.J. Watson Research Center, 1987.

[7] P. F. Brown et al., "Class-based n-gram models of natural language", Computational Linguistics 18,no.4 (1992), 467-479.

[8] M. J. F. Gales, "Maximum Likelihood Linear Transformations for HMM–Based Speech Recognition," *Computer Speech and Language* 12, pp 75–98, 1998.

[9] M. J. F. Gales, "Semi-tied Covariance Matrices for Hidden Markov Models", IEEE Trans. on Speech and Audio Processing, Vol. 7, 272-281, 1999.

[10] M. J. F. Gales, "Cluster Adaptive Training for Speech Recognition", Proceedings ICSLP 1998, pp. 1783–1786.

[11] F. Jelinek, "Statistical Methods for Speech Recognition",1997, The MIT Press, Cambridge.

[12] L. A. Liporace, "Maximum Likelihood Estimation for Multivariate Observations of Markov Sources," *IEEE Transactions Information Theory* 28, pp 729–734, 1982.

[13] Mathew A. Sieger,"Measuring and Compensating for the Effects of Speech Rate in Large Vocabulary Continuous Speech Recognition", Thesis, Carnegie Mellon University, 1995.

[14] M. Subbotin, "On the law of frequency of errors," Mathematicheskii Sbornik 31, pp 296–301, 1923

[15] Michael D. Monkowski et al., "Context Dependent Phonetic Duration Models for Decoding Conversational Speech".

[16] David Burshtein, "Robust Parametric Modelling of Durations in Hidden Markov Models", ICASSP (1995), 548-551.

[17] M. J. F. Gales, “Maximum Likelihood Linear Transformations for HMM-based Speech Recognition”, Tech. Rep., CUED/FINFENG/TR291, Cambridge Univ., 1997.

[18] J. G. Fiscus, “A post–processing system to yield reduced word error rates: recognizer output voting error reduction (rover),” *Proc. IEEE ASRU Workshop*, Santa Barabara, pp. 347–352 (1997).

[19] A. G. Richter, “Modelling of continuous speech observations,” *Advances in Speech Processing Conference,* IBM Europe Institute, 1986.

[20] R. Bakis et al., “Transcription of broadcast news shows with the IBM large vocabulary speech recognition system”, Proceedings of the Speech Recognition Workshop, pp 67-72, 1997.

[21] S. Chen et al., “IBM’s LVCSR System for Transcription of Broadcast News Used in the 1997 Hub4 English Evaluation”, Proceedings of the Speech Recognition Workshop, 1998.

[22] Chen, S S et al., “Recent Improvements to IBM’s Speech Recognition System for Automatic Transcription of Broadcast News,” *Proceedings of the Broadcast News Transcription and Understanding Workshop,* 1999.

[23] H. Beigi and S. Maes, “Speaker, channel and environment change detection”, Proceedings of the World Congress on Automation, 1998.

[24] H. Gish and N. Schmidt, “Text-independent speaker identification”, IEEE Signal Processing Magazine, pp 18-21, Oct. 1994.

[25] H. Jin, F. Kubala and R. Schwartz, “Automatic speaker clustering”, Proceedings of the Speech Recognition Workshop, pp 108-111, 1997.

[26] F. Kubala et al., “The 1996 BBN Byblos Hub-4 transcription system”, Proceedings of the Speech Recognition Workshop, pp 90-93, 1997.

[27] C. J. Legetter and P. C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density HMM’s”, Computer Speech and Language, vol. 9, no. 2, pp 171–186.

[28] M. Siegler, U. Jain, B. Ray and R. Stern, “Automatic segmentation, classification and clustering of broadcast news audio”, Proceedings of the Speech Recognition Workshop, pp 97-99, 1997.

[29] G. Schwarz, “Estimating the dimension of a model”, The Annals of Statistics, vol. 6, pp 461-464, 1978.

[30] P. Woodland, M. Gales, D. Pye and S. Young, “The Development of the 1996 HTK broadcast news transcription system”, Proceedings of the Speech Recognition Workshop, pp 73-78, 1997.

[31] Stolcke, A. and S. Omohundro. “Best-first Model Merging for Hidden Markov Model Induction.” International Computer Science Institute report TR-94-003. January 1994.

[32] R. A. Gopinath. ”Maximum Likelihood Modeling with Gaussian Distributions for Classification”, Proc. ICASSP 1998.

[33] A. Tritschler and R. A. Gopinath. ”An improved segmentation and clustering scheme using the Bayesian Information Criterion.” Proc. Eurospeech 1999, Budapest, Hungary, Sept. 1999.

[34] T. Anastasakos, et al., “A Compact Model fo r Speaker-Adaptive Training”, Proc. ICSLP-96.

[35] D. Pallet, "Overview of the 1997 DARPA Speech Recognition Workshop" , Proc. of DARPA Speech Recognition Workshop, Feb 2-5, Chantilly VA, 1997.