Speaker Adaptation Using Constrained Estimation of Gaussian Mixtures

V. Digalakis D. Rtischev L. Neumeyer

SRI International

333 Ravenswood Ave.

Menlo Park, CA 94025

EDICS SA 1.6.7

Abstract

A recent trend in automatic speech recognition systems is the use of continuous mixture-density hidden Markov models (HMMs). Despite the good recognition performance that these systems achieve on average in large vocabulary applications, there is a large variability in performance across speakers. Performance degrades dramatically when the user is radically different from the training population. A popular technique that can improve the performance and robustness of a speech recognition system is adapting speech models to the speaker, and more generally to the channel and the task. In continuous mixture-density HMMs the number of component densities is typically very large, and it may not be feasible to acquire a sufficient amount of adaptation data for robust maximum-likelihood estimates. To solve this problem, we propose a constrained estimation technique for Gaussian mixture densities. The algorithm

is evaluated on the large-vocabulary Wall Street Journal corpus for both native and nonnative speakers of American English. For nonnative speakers, the recognition error rate is approximately halved with only a small amount of adaptation data, and it approaches the speaker-independent accuracy achieved for native speakers. For native speakers, the recognition performance after adaptation improves to the accuracy of speaker-dependent systems that use 6 times as much training data.

1 Introduction

Recognition error rates ranging from 10% to 15% have recently been achieved in the 20,000-word, open-vocabulary recognition task on the Wall Street Journal (WSJ) corpus [20] using hidden Markov models (HMMs) [2, 12] with continuous-mixture observation densities [19]. However, this recognition performance is far from satisfactory for most usable large-vocabulary recognition (LVR) applications. Moreover, recognition accuracy is very sensitive to speaker variability and will degrade much more in the move from the lab to the field. Speaker-, channel-, or other task-dependent solutions require excessive collection of training data and decrease system utility and portability. A popular technique that can be used to improve the performance and robustness of a speech recognition system is adapting the speech model to the speaker, channel, and task [5, 23, 9, 15]. In this work, we consider adaptation to the speaker, although the techniques can be modified to be used at other levels.

In this paper we will present novel adaptation techniques for state-of-the-art continuous mixture-density HMMs. It has recently been shown that HMMs that use continuous-density probability distributions achieve better recognition performance than those that use discrete-density distributions [19]. After [8], we refer to a group of Gaussians that are used to form a Gaussian mixture distribution as a *genone*, to the collection of these groups as genones, and to HMM systems with an arbitrary degree of genone sharing[1] as genonic HMMs. The degree of genone sharing significantly affects recognition performance [8]. HMM systems with less sharing have typically a smaller number of Gaussians per genone and a larger total number of Gaussians than systems with fewer genones. The increase in the number of

[1] By *degree of genone sharing* we refer to the average number of distinct HMM states that share the same genone's Gaussians in their output distributions.

Gaussians is usually over-compensated for by the decrease in the number of mixture weights, and systems with less sharing have a smaller number of parameters. Hence, they are more suited to adaptation than tied-mixture HMMs (single-genone systems, with all HMM states sharing the same Gaussians in their mixture distributions).

Two families of adaptation schemes have been proposed in the past. One transforms the speaker's feature space to "match" the space of the training population [6, 18, 4]. The transformation can be applied either directly to the features, or to the speech models. The second main family of adaptation algorithms follows a Bayesian approach, where the speaker-independent information is encapsulated in the prior distributions [5, 15]. The transformation approach has the advantage of simplicity. In addition, if the number of free parameters is small, then transformation techniques adapt to the user with only a small amount of adaptation speech (quick adaptation). The Bayesian approach usually has nice asymptotic properties, that is, speaker-adaptive performance will converge to speaker-dependent performance as the amount of adaptation speech increases. However, the adaptation rate is usually slow.

For HMMs with a small degree of sharing and a large total number of Gaussians, it is impractical to expect enough adaptation data to obtain robust maximum-likelihood (ML) estimates of all the Gaussians. To deal with the problem of adapting a large number of Gaussians from small amounts of adaptation speech, we present a new algorithm for the constrained estimation of genones. The algorithm can also be viewed as estimating a transformation of the speaker-independent models by maximizing the likelihood of the adaptation data. In contrast to previous adaptation schemes based on feature transformations, our algorithm has the desirable property of being text-independent. It does not require the new speaker to record sentences with the same text recorded previously by some reference speakers, nor does it require a time warping between the new speaker's utterances and those

uttered by the reference speakers. In Bayesian adaptation techniques, the limited amount of speaker-specific data is combined with the speaker-independent models in an optimal manner. Maximum *a posteriori* (MAP) reestimation for continuous Gaussian-mixture HMMs is equivalent to linearly combining the speaker-dependent sufficient statistics with the speaker-independent priors [16]. Typically, only the Gaussians of the speaker-independent models that are most likely to have generated some of the adaptation data will be adapted to the speaker. This behavior may be problematic for continuous HMMs with a large number of Gaussians, since only a small percentage of the Gaussians will be "seen" in the adaptation data. In contrast, our adaptation scheme can adapt a Gaussian without requiring training examples of this specific Gaussian to exist in the adaptation data. By using a constrained reestimation method, our algorithm is able to extrapolate and adapt Gaussians in a genone based on data that were most likely generated by other Gaussians of the same or other neighboring genones.

This paper is organized as follows. Section 2 presents an algorithm for the constrained estimation of Gaussian mixtures based on the Expectation-Maximization (EM) algorithm. We give the solution for both the static case of a single random vector modeled by a Gaussian mixture density and the dynamic case of a vector process modeled using HMMs with Gaussian mixtures as output distributions. In Section 3 we discuss the application of the main algorithm to the speaker adaptation problem. Section 4 describes experiments and presents results on the WSJ corpus. Finally, discussion of results and conclusions appear in Section 5.

2 Constrained Estimation of Gaussian Mixtures

One speaker adaptation paradigm that fits well with the overall approach of continuous-density HMMs with shared Gaussian codebooks is to employ a transformation of the speaker-independent models to best correspond to the available adaptation data. Such a transformation can be efficiently achieved by assuming that the Gaussians in each genone of the speaker-adapted system are obtained through a transformation of the corresponding speaker-independent Gaussians. This transformation can be either unique for each genone, or shared by different genones. We choose to apply the transformation at the distribution level, rather than transforming the data directly, since we can then use the EM algorithm to estimate the transformation parameters by maximizing the likelihood of the adaptation data. The advantage of using the EM algorithm is that we can estimate the transformation from new-speaker data alone. This eliminates the need of some form of time alignment between the new-speaker data and the training- or reference-speaker data that previous transformation-based techniques needed [6, 18]. The estimation of the transformation can also be viewed as a constrained estimation of Gaussian mixtures.

2.1 Estimation of a Single Gaussian-Mixture

To better illustrate the constrained Gaussian estimation method, we first present the estimation formulae for a single Gaussian-mixture density. In Section 2.2 we extend the method for mixture densities as observation distributions in hidden Markov models. Let us consider a Gaussian mixture density of the form

$$f(x;\theta) = f(x;A,b) = \sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i) N(x; Am_i + b, AS_iA^T), \tag{1}$$

where the model parameters are $\theta = [A, b]$, N_ω is the number of mixture components, and we have the constraint that

$$\sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i) = 1. \tag{2}$$

We assume that the parameters $[m_i, S_i, i = 1, \ldots, N_\omega]$ are fixed, and that the matrices S_i are positive definite.

This model is equivalent to assuming that the random vector x is obtained through an affine transformation $x = Ay + b$ from the unobserved vector y that has a known mixture density

$$g(y) = \sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i) N(y; m_i, S_i). \tag{3}$$

ML estimation of the constrained Gaussian-mixture model is, therefore, equivalent to estimating the regression parameters A, b using only observations of the dependent variable and the knowledge of the distribution of the unobserved variable y.

As shown in [21], the EM algorithm can be used to obtain ML estimates of the parameters of a Gaussian-mixture density in the unconstrained case. The EM algorithm can also be used to estimate the model parameters $[A, b]$ in the constrained case. At each EM iteration, the new parameter estimates are obtained by maximizing the auxiliary function [7]

$$\theta_n = \arg\max_\theta E\{\log \mathrm{p}(\mathcal{X}, \Omega|\theta)|\mathcal{X}, \theta_o\}, \tag{4}$$

where $\theta_o = [A_o, b_o]$ are the previous parameter estimates, $\mathcal{X}$ denotes the collection of observed samples x, and Ω denotes the collection of the corresponding unobserved mixture indices ω_i.

Each iteration of the EM algorithm involves an expectation (E-step) and a maximization step (M-step). In the Appendix we show that the E-step involves the computation of the sufficient statistics

$$\bar{\mu}_i \quad = \quad \frac{1}{n_i} \sum_x \mathrm{p}(\omega_i|A_o, b_o, x) x \tag{5}$$

$$\bar{\Sigma}_i = \frac{1}{n_i}\sum_x \mathrm{p}(\omega_i|A_o, b_o, x)(x-\bar{\mu}_i)(x-\bar{\mu}_i)^T \tag{6}$$

$$n_i = \sum_x \mathrm{p}(\omega_i|A_o, b_o, x), \tag{7}$$

where the posterior probabilities can be computed using Bayes' rule

$$\mathrm{p}(\omega_i|A_o, b_o, x) = \frac{\mathrm{p}(\omega_i)N(x; A_o m_i + b_o, A_o S_i A_o^T)}{\sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i)N(x; A_o m_i + b_o, A_o S_i A_o^T)}. \tag{8}$$

For the one-dimensional case, and therefore for the case of diagonal covariances and a diagonal scaling matrix A, the quantities $S_i = s_i^2, A = a, \bar{\Sigma}_i = \bar{\sigma}_i^2$ and $\bar{\mu}_i, m, b$ are scalars. In this case, the M-step is equivalent to solving the following quadratic equation (see Appendix):

$$\Big(\sum_{i=1}^{N_\omega} n_i\Big)a^2 - \Big(\sum_{i=1}^{N_\omega} \frac{n_i}{s_i^2}\Big)b^2 - \Big(\sum_{i=1}^{N_\omega} \frac{n_i m_i}{s_i^2}\Big)ab + \Big(\sum_{i=1}^{N_\omega} \frac{n_i \bar{\mu}_i m_i}{s_i^2}\Big)a + \Big(2\sum_{i=1}^{N_\omega} \frac{n_i \bar{\mu}_i}{s_i^2}\Big)b - \Big(\sum_{i=1}^{N_\omega} n_i \frac{\bar{\mu}_i^2 + \bar{\sigma}_i^2}{s_i^2}\Big) = 0 \tag{9}$$

where the offset b is given by

$$b = \Big(\sum_{i=1}^{N_\omega} \frac{n_i \bar{\mu}_i}{s_i^2} - a\sum_{i=1}^{N_\omega} \frac{n_i m_i}{s_i^2}\Big) \Big/ \Big(\sum_{i=1}^{N_\omega} \frac{n_i}{s_i^2}\Big). \tag{10}$$

It is straightforward to verify that this equation has real roots. For the general multidimensional case—that is, when the covariances and the scaling matrix A are not diagonal—the M-step is equivalent to solving a system of second order equations. Iterative schemes may be used in the general case.

2.2 Estimation of a Gaussian Mixture Density in HMMs

The constrained estimation of Gaussian mixtures can be easily extended for the dynamic case of time-varying processes with an underlying discrete Markovian state. Specifically, consider the finite-state process $[s_t, t = 1, \ldots, T]$, which can be modeled as a first-order Markov chain with transition probabilities $a_{ij} = \mathrm{p}(s_t = j|s_{t-1} = i)$. This state process can generate an observed process $[x_t]$ through a stochastic mapping $\mathrm{p}(x_t|s_t)$, and the overall

model for the process $[x_t]$ is a hidden Markov model. In the reestimation formulae for HMMs with Gaussian mixture output distributions of the form

$$\mathrm{p}(x_t|s_t) = \sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i|s_t) N(x_t; A(g)m_i(g) + b(g), A(g)S_i(g)A^T(g)), \tag{11}$$

g is the Gaussian codebook (or genone) index. Thus, we assume that we have a collection of genones indexed by $g = 1, \ldots, N_g$, and that the mapping from HMM state s_t to genone is $g = \gamma(s_t)$. The inverse image $\gamma^{-1}(g)$ is the set of HMM states that map to the same genone (i.e., the set of HMM states that share the same mixture components). As in the static case, we assume that the parameters $m_i(g), S_i(g), i = 1, \ldots, N_\omega$ are fixed, the matrices $S_i(g)$ are positive definite, and the free parameters in the mixtures are the transformation parameters $A(g), b(g)$ which, for simplicity, are assumed to be genone-dependent.

The EM algorithm can be used to estimate the parameters of this model. The unobserved variables are the HMM state and the mixture index, and the EM algorithm in this case takes the form of the well-known Baum-Welch algorithm [3]. The formulae for the conventional reestimation of HMMs with Gaussian mixture densities can be derived by applying the Baum-Welch algorithm; see, for example, [13]. In our case, since we constrain the estimation of the Gaussians, the reestimation formulae are different, and the training procedure using the Baum-Welch algorithm is as summarized below.

1. Initialize all transformations with $A_0(g) = I, b_0(g) = 0, g = 1, \ldots, N_g$. Set $k = 0$.

2. **E-step:** Perform one iteration of the forward-backward algorithm on the speech data, using Gaussians transformed with the current value of the transformations $\theta_k(g) = [A_k(g), b_k(g)]$. For all component gaussians and all genones g collect the sufficient statistics

$$\bar{\mu}_i(g) = \frac{1}{n_i(g)} \sum_t \sum_{s_t \in \gamma^{-1}(g)} \rho(s_t)\phi_{it}(g)x_t \tag{12}$$

$$\bar{\Sigma}_i(g) = \frac{1}{n_i(g)} \sum_t \sum_{s_t \in \gamma^{-1}(g)} \rho(s_t)\phi_{it}(g)(x_t - \bar{\mu}_i(g))(x_t - \bar{\mu}_i(g))^T \tag{13}$$

$$n_i(g) = \sum_t \sum_{s_t \in \gamma^{-1}(g)} \rho(s_t)\phi_{it}(g), \tag{14}$$

where $\rho(s_t) = \mathrm{p}(s_t|\mathcal{X}, \lambda_k)$ is the probability of being at state s_t at time t given $\mathcal{X}$ and the current HMM parameters λ_k, and $\phi_{it}(g)$ is the posterior probability

$$\phi_{it}(g) = \mathrm{p}(\omega_i(g)|A_k(g), b_k(g), x_t, s_t). \tag{15}$$

3. **M-step:** Compute the new transformation parameters $[A_{k+1}(g), b_{k+1}(g)]$ using the estimation formulae (9), (10).

4. If another iteration, goto (2).

3 Application to Speaker Adaptation

3.1 Adaptation of Gaussian Codebooks

For continuous mixture-density HMMs with a large number of component mixtures it is impractical to assume that there are enough adaptation data available for independent reestimation of all the component densities. The constrained estimation that we have presented can overcome this problem, since all the components within a mixture (or a group of mixtures, if there is tying of transformations) are transformed jointly. To see how this method can be applied for adaptation, we assume that the speaker-independent (SI) HMM model for the SI vector process $[y_t]$ has observation densities of the form

$$p_{SI}(y_t|s_t) = \sum_{i=1}^{N_\omega} \mathrm{P}(\omega_i|s_t) N(y_t; m_i(g), S_i(g)). \tag{16}$$

Adaptation of this system can be achieved by jointly transforming all the Gaussians of each genone. Specifically, we assume that, given the genone index of the HMM state s_t, the speaker-dependent vector process $[x_t]$ can be obtained by the underlying process $[y_t]$ through the transformation $x_t = A(g)y_t + b(g)$. In this case, the speaker-adapted (SA) observation densities have the form

$$p_{SA}(x_t|s_t) = \sum_{i=1}^{N_\omega} \mathrm{P}(\omega_i|s_t) N(x_t; A(g)m_i(g) + b(g), A(g)S_i(g)A^T(g)), \tag{17}$$

and only the transformation parameters $A(g), b(g), g = 1, \ldots, N_g$ need to be estimated during adaptation. We chose groups of affine transformations to model the underlying relationship between the speaker-independent and the speaker-adapted densities for two reasons: first, constraining the reestimation of Gaussian mixtures with affine transformations results to a mathematically tractable problem; and second, by increasing the number of transformations we can achieve a good approximation of any underlying relationship. Although we presented

in this paper the reestimation formulae for a full-rank transformation, in our experiments we used independent constraints, that is diagonal covariances and scaling matrices.

The above algorithm can also be modified to asymptotically approach speaker-dependent (SD) training as the amount of adaptation speech is increased. We can achieve this by setting a threshold and reestimating without constraints all individual Gaussians for which the number of samples assigned to them is larger than the threshold. Hence, all Gaussians with a sufficiently large amount of adaptation speech are reestimated independently, whereas Gaussians with little or no adaptation data are adapted in groups. In addition, if the total amount of adaptation data for a particular genone is less than a prespecified threshold, then an identity transformation is used for all of its Gaussians.

Since our Gaussian adaptation algorithm is an instance of the Baum-Welch algorithm for HMMs with constrained mixture densities, it can be implemented efficiently. Specifically, the sufficient statistics (12) through (14) are the same as in the case of unconstrained mixture densities. Hence, the E-step at each iteration of the adaptation algorithm requires the computation and storage of these statistics and is equivalent to the E-step of the Baum-Welch algorithm for unconstrained mixture densities. The computational requirements of the M-step are very small compared to the E-step.

3.2 Adaptation of Mixture Weights

The constrained estimation algorithm that we described in the previous sections can be used to adapt the component densities of the observation distributions. Another set of parameters in a continuous-mixture HMM speech recognizer is comprised by the mixture weights $\mathrm{p}(\omega_i|s_t)$. When there is a high degree of sharing of the mixture components among different HMM states—that is, when the number of genones N_g is small—then the distri-

butions corresponding to different HMM states are mainly distinguished by the different mixture weights. In HMMs with less sharing, as N_g increases, there is a shift in focus and the discrimination between different states is mainly achieved using the component densities. Hence, the significance of adapting the mixture weights varies, depending on the type of sharing. Since systems with a small degree of sharing usually perform better, adaptation of the Gaussians may have a greater effect on recognition performance. Nevertheless, it may still prove beneficial to incorporate in the adaptation scheme some form of adaptation of the mixture weights.

The technique that we chose to use can be characterized as "pseudo-Bayesian". Specifically, after adapting the component Gaussians as described in Section 3.1, an additional pass through the adaptation data is performed using the forward-backward algorithm. The SD counts for the mixture weights are accumulated, and linearly combined with the SI forward-backward counts, in a fashion similar to the one reported in [10]. The weighting factor that is used determines the relative prominence given to the adaptation data. The algorithm can also be viewed as a pseudo-Bayesian adaptation scheme, where the relative contribution of the SI prior knowledge and the SD adaptation data is determined experimentally.

4 Experiments

We evaluated our adaptation algorithms on the large-vocabulary Wall Street Journal corpus [20]. Experiments were carried out using SRI's DECIPHER^{TM} speech recognition system configured with a six-feature front end that outputs 12 cepstral coefficients ($c_1 - c_{12}$), cepstral energy (c_0), and their first- and second-order differences. The cepstral features are computed from an FFT filterbank, and subsequent cepstral-mean normalization on a sentence basis is performed. We used genonic hidden Markov models with an arbitrary degree

of Gaussian sharing across different HMM states as described in [8]. For fast experimentation, we used the progressive search framework [17]: an initial, speaker-independent recognizer with a bigram language model outputs word lattices for all the utterances in the test set. These word lattices are then rescored using speaker-dependent or speaker-adapted models and the same bigram language model. We performed two series of experiments, on native and nonnative speakers of American English, respectively. All experiments were performed on the 5,000-word closed-vocabulary task, and are described below.

4.1 Adaptation to Native Speakers

To compare SI, SD and SA recognition performance on native speakers, we performed an initial study of our adaptation algorithms on the phase-0 WSJ corpus. We used phonetically-tied mixture HMM systems, with all allophones of the same context-independent phone sharing the same mixture components, that is, we used systems with one genone per phone. Speaker-independent systems were trained on 3,500 sentences from 42 male speakers. The different cepstral features were modeled as independent observation streams, and each codebook used 50 Gaussians for the vector features and 15 Gaussians for the scalar (energy) features. There was a total of 6,300 phonetic models, each with three states. The number of distinct output distributions was clustered down to 6,300 (a 3-fold reduction) using state-based clustering [11], since a more compact system with fewer parameters is better suited for adaptation. The performance of the adaptation algorithm was evaluated on 100 sentences from each of six male speakers (001, 00b, 00c, 00d, 400, and 431) for varying amounts of training/adaptation sentences. The SI word error rate for these speakers was 15.08% before clustering the distributions, including deletions and insertions. Clustering degraded slightly the SI performance to 15.51% word error. Since we used a phonetically-tied mixture system with 40 phonetic classes, the total number of Gaussians in the system was only 2000 and

600 for each vector and scalar feature, respectively. We were, therefore, able to train a speaker-dependent system for each one of the six speakers using 650 utterances, and we found that the SD error rate was 11.51%. We then tested the adaptation algorithm using a small amount of adaptation data (40 phonetically-balanced utterances common across all speakers), and the word error rate after adaptation was 13.60%. Thus, with 40 adaptation sentences, 60% of the gap between SI and SD performance was overcome.

We then evaluated the SA system performance for varying amounts of adaptation data, using three of the speakers. The results are summarized in Figure 1. With 100 adaptation sentences, the adaptation scheme achieves the performance of a speaker-dependent system that used 6 times as much speaker-specific training data. When all the SD training data are used as adaptation data, the SA system achieves a 50% reduction in error rate over the SI system and a 25% reduction over the SD system.

It is difficult to compare our work to other adaptation schemes that have appeared in the literature. The results are usually confounded by differences in:

- the task complexity. This includes vocabulary size, use of a strict language model, noise conditions, etc.
- the type of recognition system and its baseline accuracy. Systems that already exhibit a good SI performance may show small improvement due to adaptation
- the fluency of the speakers and the test-sample size. As we will see in the following section, adaptation helps nonnative speakers significantly more than native speakers.

In order to overcome some of these problems and compare our algorithm to previous work, we implemented the adaptation algorithm described in [22]. This algorithm is only suitable for tied-mixture systems: adaptation of the Gaussians is achieved using uncon-

strained Baum-Welch reestimation and there is no mixture-weight adaptation. We built an SI tied-mixture system and found that the SI and 40-sentence SA word error rates on the six-speaker test set were 17.0% and 16.1%, respectively. Both of these numbers are higher than the 15.5% and 13.6% word error rates that we observed using SI phonetically-tied mixtures and our adaptation algorithm.

Because of the reasons we mentioned above, we can only make qualitative comments in comparing our algorithm to previous work by others. In [16], Lee and Gauvain obtained similar SD and SA recognition performance (3.5% word error rate) with 600 sentences on the 1,000-word ARPA Resource Management (RM) task using context-independent models. Our adaptation algorithm achieved 25% lower error than SD training when 650 WSJ sentences were used. With 40 adaptation sentences, their method reduced the SI word error rate by 33% (from 6.3% to 4.2%). In our case we observed a 12% reduction. However, both of these differences may be attributed to the different domains, the amount of initial SI training data and the quality of the SI models.

Huang and Lee [10] also reported adaptation results on the RM task. They used the simple Gaussian reestimation scheme proposed by Rtischev [22] and a "pseudo-Bayesian" adaptation method for the mixture weights that is similar to the one we used in our work. On a different test set from the one used by Lee and Gauvain, they reported a 4.3% SI word error rate and a 2.6% SD word error rate using 600 SD training sentences. Their SA results were 3.6%, 2.5% and 2.4% using 40, 300 and 600 adaptation sentences, respectively. Their error rates are, in general, lower than the ones in [16]. As a consequence, Huang and Lee's error-rate reduction using 40 adaptation sentences is smaller (16%) than Lee and Gauvain's, and is comparable to ours. Also, the Huang-Lee method achieves 600-sentence SD performance after 300 adaptation sentences, and the 600-sentence SA error rate is 8% less than the corresponding SD error rate. In our case, we achieved 650-sentence SD performance

after 100 adaptation sentences and our 650-sentence SA error rate is 25% lower than the corresponding SD error rate.

4.2 Adaptation to Nonnative Speakers

Speaker adaptation becomes a very important technology for outlier speakers, since the SI error rate is too high for any practical application[2]. In testing the adaptation algorithm on the "spoke 3" task of the phase-1 Wall Street Journal corpus [14], we focused on improving recognition performance for nonnative speakers of American English using adaptation. Since the phase-1 corpus was available during this series of experiments, the SI systems were built using 17,000 training utterances from 140 male speakers. To reduce computing requirements we tuned the algorithm using the five male speakers in the phase-1 WSJ development data set. A bigram language model was used in all of our experiments. The evaluation data set was run only once at the end of the development phase. The data set includes 40 test sentences and 40 phonetically balanced adaptation sentences per speaker. The speakers were selected according to their fluency in English, covering strong to light accents.

We first tested four different systems to determine the optimal degree of Gaussian sharing for this task. All of the systems used 11,932 context-dependent phonetic models, each with three states. Context dependency was modeled only within words, since we had found in preliminary experiments that modeling coarticulation across word boundaries does not improve recognition performance for nonnative speakers. The numbers of genones used in these systems were 40 (1 genone per phone), 200, 500, and 950. Each genone consisted of a mixture of 32 Gaussian distributions. The SI and SA performance is shown in Table 1. The adaptation was applied sequentially to the Gaussian distributions and the mixture

[2]This was an additional motivation for all three authors of this paper, who are nonnative speakers of American English. Two of the authors are actually included in the test sets used in this section's experiments.

weights. In general, an increase in the number of genones also increases the computational requirements during recognition because of the larger number of Gaussian likelihoods that need to be evaluated. However, these Gaussian evaluations may be sped up using methods like clustering and vector quantization [8].

In genonic HMMs, an arbitrary degree of mixture tying across different HMM states can be selected through an agglomerative clustering procedure [8]. If the degree of tying is small, and consequently the number of genones is large (as in the 500- and 950-genone systems in Table 1), then a large number of transformations may have to be estimated during adaptation. We can overcome this problem by using tying of the transformations across different genones, and the agglomerative clustering scheme used for the genone construction is very suitable for this. Each node in the tree that is generated during the clustering procedure corresponds to a set of states, with the leaves of the tree corresponding to single HMM states. The degree of tying used in a particular system can be represented by a cut through the tree. The location of the cut is determined by the stopping criterion of the agglomerative clustering. Thus, if we want to use a smaller number of transformations than the number of genones in the system, we can somewhat relax the stopping criterion (i.e., cluster more aggressively) and determine a second cut, at a higher level through the tree. All nodes of the original cut (i.e., all genones) that fall under the same node of the new cut can share the same transformation. The third column in Table 1 indicates the number of transformations used in reestimating the Gaussian distributions. In the first two systems we used one transformation per genone. In the remaining two systems with large numbers of genones, we grouped the transformations in order to reduce the number of parameters to be estimated.

The SI word error rates for the various systems were similar, ranging from 28.7% to 30.1%. By using tying of the transformations during adaptation for the 950- and 500-genone

systems and reducing the number of transformations from 950 and 500 to 200, the SA error rates were reduced from 17.7% and 16.6% to 15.8% and 15.1%, respectively. The SA error rate of 15.1% was the lowest overall for all the systems that we examined, and the average improvement due to the adaptation algorithm for the five speakers was 47%. To evaluate the relative contribution of the two stages of our adaptation scheme, we evaluated the SA error rate for our best system with the mixture-weight adaptation disabled. We found that by adapting the Gaussian codebooks only using the constrained estimation method, the SA word error rate was 15.6%. Hence, for continuous HMMs most of the performance gain during adaptation is achieved by adapting the Gaussian codebooks. Table 2 shows the results for the November 1993 ARPA evaluation set [19] on the 500-genone system. In this case the improvement is 27%. The difference between the development and evaluation test sets can be attributed to the large variability that is inherent in these outlier speakers and the relatively small test-set size. To further evaluate the performance of our algorithm, we tested it on the full November 1993 evaluation set, including all 10 male and female speakers. The SI word error rate was 23.1%, and was reduced to 14.8% using our adaptation algorithm. This result is comparable to that obtained by Kubala in the official November 1993 evaluation using a trigram language model [19].

To compare the nonnative performance before and after adaptation to that of native speakers, we evaluated the same four systems on the same speakers that we used in Section 4.1. The results are summarized in Table 3. There we see that the SI performance of the more detailed systems (with a larger number of Gaussian distributions) is significantly better than that of the less detailed ones. This is an important difference from the nonnative results. A plausible explanation for the nonnative case is that the additional detail of the more continuous systems is not needed if the speakers are different from the training population. We also observe that for natives the SA error rate using 40 utterances is only 7% less than

the SI one, as opposed to the 30% to 50% improvement that we observed for nonnatives. Moreover, the improvement is less than the 12% decrease in word error that was observed for the native speakers in the experiments with the phase-0 WSJ corpus, and is not uniform across speakers. Since the phase-1 WSJ corpus has 5 times more training data than the phase-0 corpus, we can conclude that, when a large amount of SI training data is available, adaptation is not nearly as effective for speakers drawn from a population that matches the training data as it is for outlier speakers.

The SI and SA word-error rates for the best systems and for both native and nonnative speakers are summarized in Table 4. The SI word error rate for nonnative speakers is 2.5 to 3 times less than that of native speakers. However, after adapting with 40 adaptation utterances, the nonnative SA error rate is approximately a factor of 1.5 higher than that of native speakers.

5 Summary

We have presented a new algorithm for the maximum-likelihood (ML) estimation of a mixture of Gaussians subject to the constraint that all means and covariances are obtained through a transformation (that needs to be estimated) from a fixed set of component densities. This constrained estimation method is well suited to the speaker adaptation problem for continuous mixture-density HMMs with a large number of component densities that are hard to estimate in an unconstrained fashion from a small amount of adaptation data.

We tested our algorithm on the large-vocabulary WSJ corpus on both native and nonnative speakers of American English, and on a variety of recognition systems. We found that for native speakers the recognition performance after adaptation is similar to that of

speaker-dependent systems that use 6 times as much training data. With small amounts of adaptation data (40 utterances with an average length of 10 seconds) the decrease in word-error rate for native speakers is approximately 7% and is much larger for nonnative speakers, ranging from 30% to 50%. This is a very important result, since the speaker-independent word-error rates for outlier speakers, like nonnative speakers, can be 2.5 to 3 times as high as those of native speakers. With speaker adaptation, outlier and nonnative speakers can use automatic speech recognition at performance levels similar to those of native speakers. Thus, the algorithm that we propose can significantly increase the usability of continuous mixture-density HMM systems. Moreover, we used the WSJ database and our results can serve as a benchmark to other researchers that want to evaluate their nonnative-speaker adaptation techniques on the same data.

We also studied the relationship between adaptation behavior and degree of mixture sharing in continuous HMM systems. We found that, with a large amount of speaker-independent training, more continuous systems with a large number of Gaussians perform better on typical native speakers in both their speaker-independent and speaker-adapted modes. However, the situation is different for atypical, nonnative speakers. For those, increasing the detail in the modeling of context dependencies is not as beneficial, since the nonnative speakers are less likely to follow the typical coarticulation patterns observed in native speakers. The result is that more compact systems actually exhibit better adaptation performance because there are fewer parameters to adapt.

Since the results of this study are very encouraging, we are currently investigating methods to extend our adaptation algorithm to work in an unsupervised manner, that is, when the prompting text is not available for adaptation.

APPENDIX: Derivation of the Expectation and Maximization Steps

To apply the Expectation-maximization (EM) algorithm to the estimation of a Gaussian mixture, we can rewrite the auxiliary function as

$$E\{\log \mathrm{p}(\mathcal{X}, \Omega|\theta)|\mathcal{X}, \theta_o\} = \sum_x \sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i|x, \theta_o)[\log \mathrm{p}(x|\omega_i, \theta) + \log \mathrm{p}(\omega_i|\theta)] \tag{18}$$

Since the parameters θ consist of the transformation parameters $[A, b]$, the second term in the summation does not depend on θ, and hence at each EM iteration we need to maximize the first term only.

It is well known that the joint log-likelihood of a collection of samples $\mathcal{X}$ drawn independently from a multivariate normal distribution with mean μ and covariance Σ can be expressed as [1]

$$\log \mathrm{p}(\mathcal{X}) = -\frac{n}{2}\log|\Sigma| - \frac{n}{2}(\mu - \bar{\mu})^T\Sigma^{-1}(\mu - \bar{\mu}) - \frac{n}{2}\mathrm{trace}\{\Sigma^{-1}\bar{\Sigma}\} \tag{19}$$

where $\bar{\mu}, \bar{\Sigma}$ are the sample mean and covariance, respectively, and n is the number of samples. A similar expression can be derived for the first term of the expected log-likelihood in (18). We first note that this expectation can be written

$$\begin{aligned}
\mathcal{L}(\theta; \theta_o) &= E\{\log \mathrm{p}(\mathcal{X}|\Omega, \theta)|\mathcal{X}, \theta_o\} && (20)\\
&= \sum_x \sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i|x, \theta_o)\left[-\frac{1}{2}\log|\Sigma_i| - \frac{1}{2}(x - \mu_i)^T\Sigma_i^{-1}(x - \mu_i)\right] && (21)\\
&= \sum_x \sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i|x, \theta_o)\Big[-\frac{1}{2}\log|\Sigma_i| + \frac{1}{2}x^T\Sigma_i^{-1}\mu_i + \frac{1}{2}\mu_i^T\Sigma_i^{-1}x \\
&\quad -\frac{1}{2}x^T\Sigma_i^{-1}x - \frac{1}{2}\mu_i^T\Sigma_i^{-1}\mu_i\Big], && (22)
\end{aligned}$$

where the means and covariances are constrained $\mu_i = Am_i + b, \Sigma_i = AS_iA^T$. By expanding

the summation above, we can write

$$\begin{aligned}\mathcal{L}(\theta;\theta_o) &= \sum_{i=1}^{N_\omega}\Big[-\frac{1}{2}\{\sum_x \mathrm{p}(\omega_i|x,\theta_o)\}\log|\Sigma_i| + \frac{1}{2}\{\sum_x \mathrm{p}(\omega_i|x,\theta_o)x^T\}\Sigma_i^{-1}\mu_i \\ &\quad +\frac{1}{2}\mu_i^T\Sigma_i^{-1}\{\sum_x \mathrm{p}(\omega_i|x,\theta_o)x\} - \frac{1}{2}\sum_x \mathrm{p}(\omega_i|x,\theta_o)x^T\Sigma_i^{-1}x \\ &\quad -\frac{1}{2}\{\sum_x \mathrm{p}(\omega_i|x,\theta_o)\}\mu_i^T\Sigma_i^{-1}\mu_i\Big]. \end{aligned} \tag{23}$$

We can define the sufficient statistics

$$n_i = \sum_x \mathrm{p}(\omega_i|A_o,b_o,x) \tag{24}$$

$$\bar{\mu}_i = \frac{1}{n_i}\sum_x \mathrm{p}(\omega_i|A_o,b_o,x)x \tag{25}$$

$$\bar{\Sigma}_i = \frac{1}{n_i}\sum_x \mathrm{p}(\omega_i|A_o,b_o,x)(x-\bar{\mu}_i)(x-\bar{\mu}_i)^T, \tag{26}$$

and rewrite equation (23) above as

$$\begin{aligned}\mathcal{L}(\theta;\theta_o) &= \sum_{i=1}^{N_\omega}\Big[-\frac{n_i}{2}\log|\Sigma_i| - \frac{n_i}{2}(\mu_i-\bar{\mu}_i)^T\Sigma_i^{-1}(\mu_i-\bar{\mu}_i) + \frac{n_i}{2}\bar{\mu}_i^T\Sigma_i^{-1}\bar{\mu}_i \\ &\quad -\frac{1}{2}\sum_x \mathrm{p}(\omega_i|x,\theta_o)x^T\Sigma_i^{-1}x\Big] \qquad (27) \\ &= \sum_{i=1}^{N_\omega}\Big[-\frac{n_i}{2}\log|\Sigma_i| - \frac{n_i}{2}(\mu_i-\bar{\mu}_i)^T\Sigma_i^{-1}(\mu_i-\bar{\mu}_i) \\ &\quad -\frac{1}{2}\mathrm{trace}\{\Sigma_i^{-1}[\sum_x \mathrm{p}(\omega_i|x,\theta_o)xx^T - n_i\bar{\mu}_i\bar{\mu}_i^T]\}\Big] \qquad (28) \\ &= -\sum_{i=1}^{N_\omega}\frac{n_i}{2}\Big[\log|\Sigma_i| + (\mu_i-\bar{\mu}_i)^T\Sigma_i^{-1}(\mu_i-\bar{\mu}_i) + \mathrm{trace}\{\Sigma_i^{-1}\bar{\Sigma}_i\}\Big], \qquad (29)\end{aligned}$$

where in the second equation above we used the matrix identity $x^TAx = \mathrm{trace}\{Axx^T\}$ for a matrix A and a vector x, and in the third equation we used the definition of the statistic $\bar{\Sigma}_i$. The equations for the computation of the sufficient statistics comprise the E-step of the algorithm, and are summarized in (5) through (7).

To derive the M-step of the algorithm, we first rewrite Equation (29) using the transformation parameters

$$\mathcal{L}(\theta;\theta_o) = -\sum_{i=1}^{N_\omega}\frac{n_i}{2}\Big[\log|S_i| + \log|A|^2 + (A^{-1}\bar{\mu}_i - m_i - A^{-1}b)^TS_i^{-1}(A^{-1}\bar{\mu}_i - m_i - A^{-1}b)$$

$$+\text{trace}\{A^{-T}S_i^{-1}A^{-1}\bar{\Sigma}_i\}\Big], \tag{30}$$

where we have assumed that the transformation matrix A has full rank. By taking the gradient of $\mathcal{L}(\theta;\theta_o)$ with respect to the transformation parameters A, b we find the following system of equations:

$$\sum_{i=1}^{N_\omega} n_i \Big\{ A - S_i^{-1}\Big[A^{-1}(\bar{\mu}_i - b) - m_i\Big](\bar{\mu}_i - b)^T - S_i^{-1}A^{-1}\bar{\Sigma}_i \Big\} = 0 \tag{31}$$

$$b = \Big[\sum_{i=1}^{N_\omega} n_i A^{-T}S_i^{-1}A^{-1}\Big]^{-1}\Big[\sum_{i=1}^{N_\omega} n_i A^{-T}S_i^{-1}A^{-1}(\bar{\mu}_i - Am_i)\Big]. \tag{32}$$

Under the assumption of diagonal covariance matrices and diagonal transformation matrices, the multidimensional case is equivalent to a set of one-dimensional problems that can be solved independently. The auxiliary function can be written in this case as

$$\mathcal{L}(\theta;\theta_o) = -\sum_{i=1}^{N_\omega} \frac{n_i}{2}[\log s_i^2 + \log a^2 + \frac{(\bar{\mu}_i - am_i - b)^2}{a^2 s_i^2} + \frac{\bar{\sigma}_i^2}{a^2 s_i^2}]. \tag{33}$$

By maximizing this quantity with respect to the transformation parameters a, b we can easily derive equations (9), (10).

Acknowledgments

We gratefully acknowledge support for this work from ARPA through Office of Naval Research Contracts N00014-93-C-0142 and N00014-92-C-0154. The Government has certain rights in this material. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Government funding agencies. We would also like to thank our colleagues Mike Cohen, Hy Murveit and Mitch Weintraub for their comments that improved the quality of this manuscript.

References

[1] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis,* 2nd Edition, Wiley, New York, 1984.

[2] L. R. Bahl, F. Jelinek and R. L. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-5(2), pp. 179–190, March 1983.

[3] L. E. Baum, T. Petrie, G. Soules and N. Weiss, "A Maximization Technique in the Statistical Analysis of Probabilistic Functions of Finite State Markov Chains," *Ann. Math. Stat.,* Vol. 41, pp. 164–171, 1970.

[4] J. Bellegarda *et al.*, "Robust Speaker Adaptation Using a Piecewise Linear Acoustic Mapping," *Proceedings ICASSP,* pp. I-445–I-448, San Fransisco, CA, 1992.

[5] P. Brown, C.-H. Lee and J. Spohrer, "Bayesian Adaptation in Speech Recognition," *Proceedings ICASSP,* pp. 761–764, Boston, MA, 1983.

[6] K. Choukri, G. Chollet and Y. Grenier, "Spectral Transformations through Canonical Correlation Analysis for Speaker Adaptation in ASR," *Proceedings ICASSP,* pp. 2659–2662, Tokyo, Japan, 1986.

[7] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood Estimation from Incomplete Data," *Journal of the Royal Statistical Society (B)*, Vol. 39, No. 1, pp. 1–38, 1977.

[8] V. Digalakis, P. Monaco and H. Murveit, "Genones: Generalized Mixture Tying in Continuous Hidden Markov Model-Based Speech Recognizers," submitted to *IEEE Trans. on Speech and Audio Processing,* June 1994.

[9] S. Furui, "Unsupervised Speaker Adaptation Method Based on Hierarchical Speaker Clustering," *Proceedings ICASSP,* pp. 286–289, Glasgow, Scotland, 1989.

[10] X. Huang and K.-F. Lee, "On Speaker-Independent, Speaker-Dependent and Speaker-Adaptive Speech Recognition," *IEEE Trans. on Speech and Audio Processing,* Vol. 1, No. 2, pp. 150–157, April 1993.

[11] M.-Y. Hwang and X. Huang, "Subphonetic Modeling with Markov States - Senone," *Proceedings ICASSP,* pp. I-33–36, San Fransisco, CA, 1992.

[12] F. Jelinek, "Continuous Speech Recognition by Statistical Methods," *IEEE Proceedings,* Vol. 64, No. 4, pp. 532–556, April 1976.

[13] B.-H. Juang, "Maximum-Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains," *AT&T Technical Journal,* Vol.64, No.6, July-August 1985.

[14] F. Kubala *et al.*, "The Hub and Spoke Paradigm for CSR Evaluation," *Proceedings of the HLT Workshop,* Princeton, NJ, March 1994.

[15] C.-H. Lee, C.-H. Lin and B.-H. Juang, "A Study on Speaker Adaptation of the Parameters of Continuous Density Hidden Markov Models," *IEEE Trans. on Acoust., Speech and Signal Proc.,* Vol. ASSP-39(4), pp. 806–814, April 1991.

[16] C.-H. Lee and J.-L. Gauvain, "Speaker Adaptation Based on MAP Estimation of HMM Parameters," *Proceedings ICASSP,* pp. II-558 – II-561, Minneapolis, Minnesota, 1993.

[17] H. Murveit, J. Butzberger, V. Digalakis and M. Weintraub, "Large-Vocabulary Dictation Using SRI's DECIPHER^{TM} Speech Recognition System: Progressive-Search Techniques," *Proceedings ICASSP,* pp. II-319 – II-322, Minneapolis, Minnesota, 1993.

[18] S. Nakamura and K. Shikano, "A Comparative Study of Spectral Mapping for Speaker Adaptation," *Proceedings ICASSP,* pp. 157–160, Albuquerque, NM, 1990.

[19] D. Pallet *et al.*, "1993 Benchmark Tests for the ARPA Spoken Language Program," *Proceedings of the HLT Workshop,* Princeton, NJ, March 1994.

[20] D. Paul and J. Baker, "The Design for the Wall Street Journal-based CSR corpus," *Proceedings of the DARPA Speech and Natural Language Workshop,* pp. 357–362, Feb. 1992.

[21] R. A. Redner and H. F. Walker, "Mixture Densities, Maximum Likelihood and the EM Algorithm," *SIAM Review*, Vol. 26, No. 2, pp. 195–239, April 1984.

[22] D. Rtischev, D. Nahamoo and M. Picheny, "Speaker Adaptation via VQ Prototype Modification," *IEEE Trans. on Speech and Audio Processing,* Vol. 2, No. 1, pp. 94–97, January 1994.

[23] R. Schwartz, Y. L. Chow and F. Kubala, "Rapid Speaker Adaptation Using a Probabilistic Spectral Mapping," *Proceedings ICASSP,* pp. 633–636, Dallas, TX, 1987.

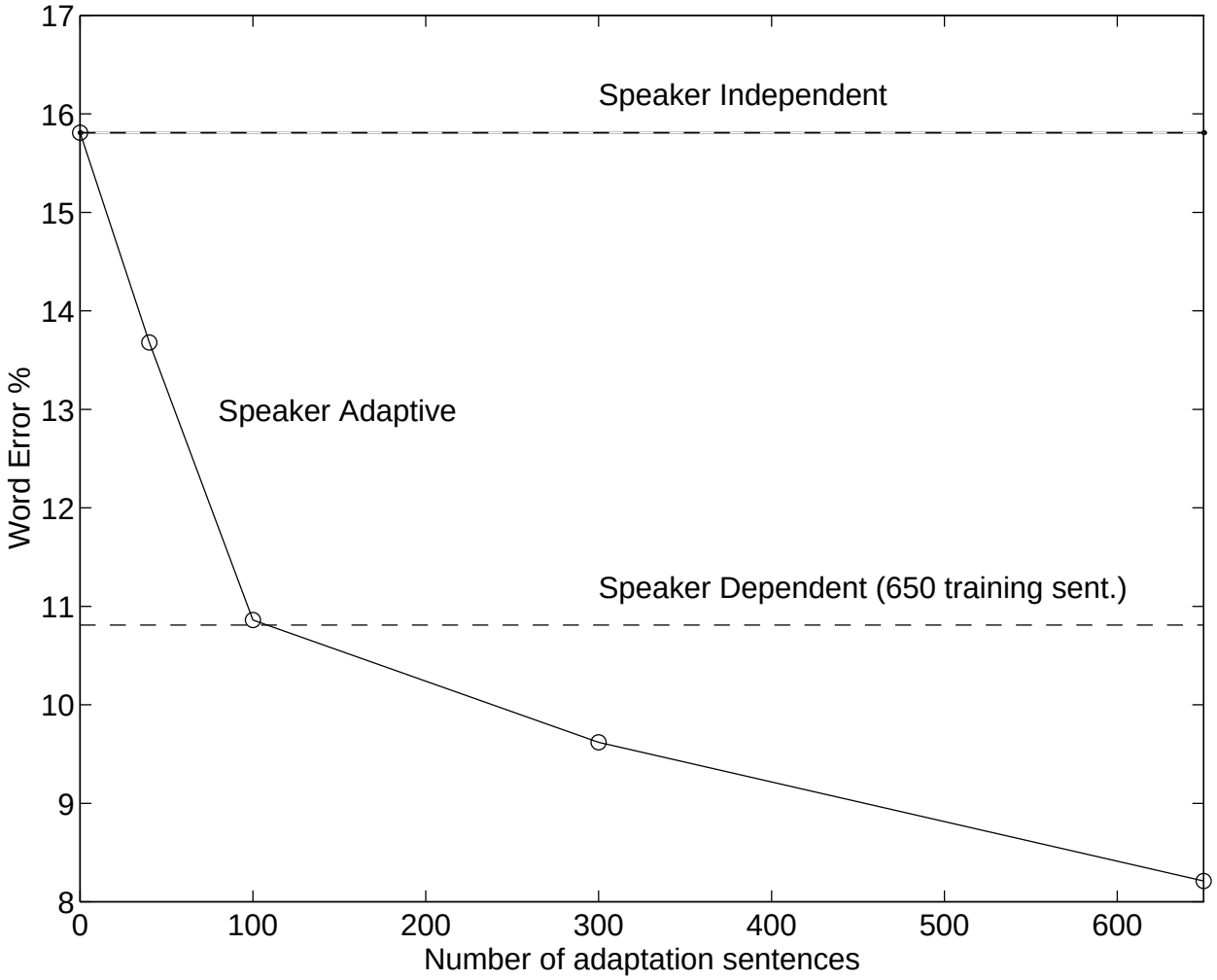

Figure 1: Speaker-independent, speaker-dependent (650 training sentences) and speaker-adaptive (varying number of sentences) word error rates for native speakers.

Speaker			4n0	4n3	4n5	4n9	4n0	AVG/SUM
Num. sentences			41	42	41	42	40	206
Num. words			719	696	664	668	678	3425
Type	Num. genones	Num. transf.						
SI	40	-	50.3	43.1	23.6	17.7	12.5	29.8
SA	40	40	24.1	18.2	17.9	12.4	9.1	16.5
SI	200	-	49.4	43.8	24.2	17.1	14.2	30.1
SA	200	200	21.4	18.7	18.4	12.0	10.5	16.2
SI	500	-	49.9	40.5	22.3	14.7	14.2	28.7
SA	500	200	20.2	15.8	16.6	12.3	10.5	15.1
SA	500	500	20.0	18.7	17.8	15.1	11.2	16.6
SI	950	-	50.5	44.7	20.5	15.3	14.4	29.5
SA	950	200	21.1	19.0	16.1	12.0	10.3	15.8
SA	950	950	24.2	21.7	18.8	13.5	9.7	17.7

Table 1: Speaker-independent (SI) and speaker-adapted (SA) word error rates for the nonnative speakers of the WSJ1 male development set for various degrees of tying and numbers of transformations.

Speaker	4nd	4ne	4nf	4ni	4nn	AVG/SUM
Num. sentences	42	42	41	41	42	208
Num. words	794	755	767	658	709	3683
SI	30.7	31.0	25.0	13.4	28.6	26.1
SA	19.0	24.5	19.7	10.2	21.0	19.1

Table 2: Word error rates for the nonnative speakers of the November 1993 WSJ1 evaluation set.

Speaker		001	00b	00c	00d	400	431	AVG/SUM
Num. sentences		50	50	50	50	50	50	300
Num. words		661	643	719	799	928	707	4457
Type	Num. genones							
SI	40	7.4	16.2	12.7	17.0	11.2	14.9	13.2
SA	40	6.7	14.8	11.3	15.3	11.0	14.1	12.3
SI	200	5.9	15.9	12.0	17.1	11.5	12.4	12.5
SA	200	6.2	16.2	13.1	13.9	10.9	12.9	12.2
SI	500	5.4	14.8	11.7	15.8	10.0	12.2	11.7
SA	500	4.8	14.8	12.0	12.8	10.0	11.3	10.9
SI	950	4.1	13.5	10.4	16.2	10.3	11.5	11.1
SA	950	3.8	13.7	11.0	12.4	9.8	10.9	10.3

Table 3: Speaker-independent (SI) and speaker-adapted (SA) word error rates for native speakers for various degrees of tying.

	SI	SA
Natives	11.1	10.3
Non natives	28.7	15.1

Table 4: Speaker-independent (SI) and speaker-adapted (SA) word error rates for native and nonnative speakers of American English.

List of Tables

1 Speaker-independent (SI) and speaker-adapted (SA) word error rates for the nonnative speakers of the WSJ1 male development set for various degrees of tying and numbers of transformations. 29

2 Word error rates for the nonnative speakers of the November 1993 WSJ1 evaluation set. 30

3 Speaker-independent (SI) and speaker-adapted (SA) word error rates for native speakers for various degrees of tying. 31

4 Speaker-independent (SI) and speaker-adapted (SA) word error rates for native and nonnative speakers of American English. 32

List of Figures

1 Speaker-independent, speaker-dependent (650 training sentences) and speaker-adaptive (varying number of sentences) word error rates for native speakers. 28