Automatic Speech Recognition Performance on a Voicemail Transcription Task

Mukund Padmanabhan, *Senior Member, IEEE*, George Saon, Jing Huang, Brian Kingsbury, and Lidia Mangu

***Abstract*—In this paper, we report on the performance of automatic speech recognition (ASR) systems on voicemail transcription. Voicemail is spontaneous telephone speech recorded over a variety of channels; consequently, it is representative of many challenging problems in speech recognition. In the course of working on this task, several algorithms were developed that focus on different components of an ASR system, including lexicon design, feature extraction, hypothesis search, and adaptation. We report the improvements provided by these techniques, as well as other standard techniques, on a voicemail test set. Although the techniques are benchmarked on voicemail test data, their scope is not restricted to this domain as they address fundamental aspects of the speech recognition process.**

***Index Terms*—Large-vocabulary conversational speech recognition, voicemail recognition.**

I. Introduction

RECENT advances in continuous speech recognition have led to high accuracy recognition of scripted speech. For example, word error rates on data from the Wall Street Journal database range from 4% to 9%, depending on whether a closed or open test vocabulary is used. However, the performance on spontaneous speech is still relatively poor. For instance, word error rates on the Switchboard corpus [1] are around 20–30% [2]. Speech encountered in real-world applications is generally spontaneous, so there is still a great deal of improvement that has to be made for speech recognition systems to be practical and usable. In this paper, we report on a number of recently developed algorithms that help to improve the accuracy of speech recognition systems and benchmark the performance of these algorithms on a voicemail transcription task.

Recognition of voicemail is an attractive area of research for a number of reasons:

- it is conversational speech transmitted over a variety of channels, including cellular telephones and speaker phones and is therefore a challenging domain for fundamental speech recognition research;
- it is distinct from other conversational speech corpora such as Switchboard [1] in that each message is a monologue and most messages are strongly goal-directed, with the speaker attempting to convey specific information;
- voicemail data is a natural testbed for work on information retrieval from speech data [3]–[5];
- there are commercial applications of reliable recognition of voicemail in the area of unified messaging.

These reasons led us to collect a large corpus of IBM voicemail data [6] for work on speech recognition and information retrieval. A subset of this data has been released to the Linguistic Data Consortium (LDC) for public distribution. Other sites are also pursuing research in this area [7]–[9].

The speech recognition process is typically cast as a search for a sequence of words that maximizes the probability for a given speech signal. The speech signal is characterized by a sequence of multidimensional feature vectors, computed by a *feature extraction* module. The *search* is constrained by a number of knowledge sources: a *lexicon* that defines the words that can be hypothesized by the search and describes each word as a sequence or network of phones (fundamental acoustic units of speech); a *language model* that assigns probabilities to hypothesized word sequences; and an *acoustic model* that models the relationship between the feature vectors and the phones. Phones are most often modeled as hidden Markov models (HMMs) and the output distributions of the HMMs are used to model the probability density of the observed feature vectors for a given phone. We report on the performance of algorithms that address several different blocks of the speech recognition process and that significantly improve the overall accuracy of speech recognition. Though the performance of some of these algorithms has been reported by varying subsets of the authors in prior conferences and workshops, the purpose of this paper is to evaluate the collective efficacy of these algorithms and to provide additional details about them.

The paper is organized as follows. In Section II we describe the different training and test sets that were used in the experiments. We also describe the basic operation of our speech recognition system. In Section III we address the problem of lexicon design and describe a data-driven method for augmenting the lexicon with compound words. In Section IV, we describe experiments related to the acoustic model, including several feature extraction and adaptation techniques. In Section V, we revisit the MAP decoding framework used in speech recognition and apply a "consensus hypothesis" processing technique to find the best hypothesis (in the minimum expected word error sense) from a word lattice. In Section VI, we describe the performance improvements obtained by adapting the features and acoustic models to the specific test speaker. In Section VII we demonstrate improvements due to system combination and in Sec-

Manuscript received September 27, 2001; revised August 1, 2002. This work was supported in part by DARPA under Grant MDA972-97-C-0012. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jerome R. Bellegarda.

M. Padmanabhan is with the Rennaisance Technologies Corporation, East Setauket, NY 11733 USA (e-mail: mukund@rentec.com).

G. Saon, J. Huang, B. Kingsbury, and L. Mangu are with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: gsaon@us.ibm.com; jghg@us.ibm.com; bedk@us.ibm.com; mangu@us.ibm.com).

Digital Object Identifier 10.1109/TSA.2002.804303

tion VIII we explore the effects of varying amounts of training data on system accuracy. We summarize our results in Section IX and conclude with a discussion of the applicability of the presented techniques to other speech recognition tasks.

II. Background

A. Training and Test Data

Our primary voicemail training set comprises 114 h of speech (approximately 1.167M words of text), divided into 12 645 messages. We will refer to this training database as T-VM1. To facilitate quicker turnaround of experiments, to provide results on publicly available data sets and to examine the effect of training data quantity on system performance, we defined four subsets of this database:

1) T-VM2, comprising 53 h of speech (6501 messages);
2) T-VM3, comprising 15 h of speech (1801 messages);
3) T-VM3b, comprising 15 h of speech (2048 messages);
4) T-VM4, comprising all of T-VM3 and T-VM3b.

T-VM3 is currently distributed by the LDC as the Voicemail Corpus Part I and T-VM3b will soon be available as the Voicemail Corpus Part II.

Test vocabularies and language models are based strictly on the available training data.[1] Given a corpus of training data, the test vocabulary is chosen to be the most frequent words covering 99% of the words in the training data. This rule leads to a test vocabulary of 19k words for systems trained on T-VM1, 11.5k for systems trained on T-VM2, 6.5k words for systems trained on T-VM3 and 8.8k words for systems trained on T-VM4. The language model is a trigram. We report results for two test sets: E-VM1, comprising 52 min of speech (105 messages) and E-VM2, comprising 35 min of speech (92 messages). E-VM2 is the subset of E-VM1 messages that are available through the LDC, with 42 messages in the Voicemail Corpus Part I and 50 messages in the Voicemail Corpus Part II. The out-of-vocabulary word rates for the different test vocabularies on the E-VM1 test set are summarized in Table I. Because the E-VM2 data is a subset of the E-VM1 data, the OOV rates in Table I are upper bounds on the OOV rates for E-VM2.

TABLE I
Out-of-Vocabulary Rates on the E-VM1 Test Set for Lexicons Derived From the Four Training Sets. The Training Sets are Ordered by Decreasing Size

Training Set	OOV Rate
T-VM1	3.75%
T-VM2	4.71%
T-VM4	5.24%
T-VM3	6.68%

B. System Description

The speech recognition system in this work uses a phonetic representation of words in the vocabulary. Each phone is modeled with a three-state, left-to-right HMM. Acoustically dissimilar variants of each state are identified using a decision network that asks questions about the acoustic context in which the state occurs and the terminals of the network correspond to the basic acoustic units we model. The acoustic context spans up to ± 5 phones around the current phone, crossing past word boundaries but ending at future word boundaries. A feature vector is extracted every 10 ms and the probability density function (pdf) of the feature vector for each acoustic unit is modeled with a mixture of Gaussians. The number of Gaussians assigned to an acoustic unit is proportional to the number of examples of that unit in a Viterbi alignment of the training data, up to some maximum number of Gaussians. The constant of proportionality and maximum number of Gaussians is set heuristically. In general, the number of units and number of Gaussians in a system increases with the amount of training data and they are kept roughly constant between systems trained on the same data set. Table XI, in an appendix, summarizes the number of acoustic units, number of Gaussians and the training data used for each system described in this paper.

In the search process, the probability densities provided by these models are not used directly. Rather, the search uses a "rank" based system in which the acoustic units are ranked on the basis of their likelihoods for a particular feature vector. Subsequently, the probability of an observation given an acoustic unit is computed by looking up the rank of the unit and converting the rank to a probability using a table lookup [10]. The hypothesis search is a modified version of the stack search known as "envelope search" [11].

The baseline feature vector is the Mel cepstrum [12] augmented with its first and second temporal derivatives (deltas). We also present experiments that used PLP cepstra [13]. Both the Mel and PLP cepstra were mean-normalized on a per-message basis. Many of the systems tested use features computed by splicing together nine cepstra (± 4 frames around the current frame) and projecting the spliced feature vector to a lower dimensional subspace using a linear transform. The derivation of this projection is described in Section IV.

III. Lexicon Design

The lexicon is a fundamental component of speech recognition systems as it defines the words that can be output by the system. If the lexicon were to contain all the words that could be uttered by a speaker and if these words could be easily disambiguated using the language and acoustic models, then high-accuracy speech recognition would be achieved. This is unfortunately not the case. However, it is generally the case that decoding errors are more common for shorter words than they are for longer words [14]. One method to capitalize on this observation is to combine groups of words that co-occur into compound words. The longer baseforms of these compound words may reduce the frequency of recognition errors. Also, cross-word coarticulation is very common in spontaneous speech. Compound words can model cross-word coarticulation. For example, the

[1]For the smaller voicemail training sets, better performance can be achieved by including data from other corpora in the lexicon and language model training [9].

phrase "going to take" could have the (compound) baseform "G AO N T AE KD".

A. Measures to Select Compound Words

While the motivation for adding compound words to the lexicon is clear, adding more baseforms to the lexicon and tokens to the language model can increase the confusability between words. Hence, the candidate pairs for compound words have to be chosen carefully. Intuitively, such a pair has to meet three requirements:

1) the pair of words must occur frequently in the training corpus;
2) the words in the pair must occur together frequently and in the context of other words less frequently;
3) the words should be coarticulated, i.e., their continuous pronunciation should be different from their concatenated isolated pronunciations.

Previously, the mutual information between a pair of words was used to select compound words [15]. This measure picks words that occur together frequently, but it fails to ensure that each word in the pair does not also occur frequently with other words. Consequently, we used a modified measure—the geometrical average of the direct and reverse bigram probabilities—to rank word pairs and select compound words [14]. The *direct bigram* probability between the words w_i and w_j is $P_f(W_{t+1} = w_j | W_t = w_i)$ and the *reverse bigram* probability between the words is $P_r(W_t = w_i | W_{t+1} = w_j)$. Both quantities can be estimated from the training corpus

$$P_f(w_j|w_i) = \frac{P(W_{t+1} = w_j, W_t = w_i)}{P(W_t = w_i)}$$
$$P_r(w_i|w_j) = \frac{P(W_{t+1} = w_j, W_t = w_i)}{P(W_{t+1} = w_j)}. \quad (1)$$

The geometrical average of the direct and the reverse bigram probabilities is

$$LM(w_i, w_j) = \sqrt{P_f(w_j|w_i) P_r(w_i|w_j)} = \frac{P(w_i, w_j)}{\sqrt{P(w_i) P(w_j)}}. \quad (2)$$

$0 \leq LM(w_i, w_j) \leq 1$ for every pair of words (w_i, w_j). A high value for $LM(w_i, w_j)$ means that both the direct and the reverse bigram probabilities are high for (w_i, w_j). In other words, the probabilities that w_i is followed by w_j and w_j is preceded by w_i are high, making the pair a good candidate for a compound word. In our implementation, we started with a lexicon containing no compound words and ran the selection process iteratively, allowing the generation of compounds of more than two words. Only word pairs occurring more than 125 times in the training data were considered as potential compounds and only the 15 pairs with the highest $LM(w_i, w_j)$ were merged in a given iteration. The procedure terminated when no additional compound words were found in an iteration.

B. Results

We report results here for a system (VM1a) trained on the T-VM2 data using the baseline, Mel cepstral features and decoding the E-VM1 data using lexicons with and without compound words. When applied to the T-VM2 data, the compound word procedure generated 70 compound words. Table II provides examples of the compound words and Table III illustrates the effect of compound words on recognition accuracy. From the examples, it appears that most of the compound words model "stock phrases" that occur frequently in voicemail data. Adding compound words based on the LM measure results in a 1.5% absolute (3.6% relative) improvement in the word error rate. All experiments described in subsequent sections used lexicons and trigram language models containing compound words found using the same procedure.

TABLE II
SAMPLE COMPOUND WORDS. UNDERLINED SETS OF WORDS IN THE EXAMPLES ARE ALSO COMPOUND WORDS

Sample compound words
AREA-CODE
E-MAIL
THANK-YOU-VERY-MUCH
THANKS-A-LOT
TAKE-CARE
LET-ME-KNOW
AS-SOON-AS
TALK-TO-YOU-LATER-BYE
PLEASE-GIVE-ME-A-CALL

TABLE III
WORD ERROR RATES (WER) FOR THE VM1A SYSTEM ON THE E-VM1 DATA WITH AND WITHOUT COMPOUND WORDS

no compound words	with compound words
42.0%	40.5%

IV. FEATURE EXTRACTION

The most commonly used feature extraction schemes extract a multi-dimensional feature vector from the sampled speech signal at a uniform frame rate (typically every 10 ms). The feature extraction procedure often mimics the approximately constant-Q frequency analysis and compressive, nonlinear response to signal energy observed in mammalian auditory systems. The d-dimensional feature vectors contain information about the local spectral characteristics of the speech signal, but they do not contain any information related to the trajectory of the spectral characteristics over time. Inclusion of trajectory information improves the discriminability of the acoustic units and is more consistent with the HMM conditional independence assumption. Trajectory information is frequently incorporated into the feature vector by concatenating the first- and second-order temporal derivatives of the features to the static features, producing a $3d$-dimensional feature vector. Alternatively, a linear discriminant transformation can be used. In this approach, the d-dimensional features from several adjacent frames (typically nine) are concatenated to form a $9d$-dimensional feature vector. The dimensionality of the

feature vector is then reduced via a linear projection designed to maximally separate the phonetic classes. In this section, we describe a process for computing a linear transformation on the features that extracts the most relevant information in the spectral features, as well as their trajectories, to separate the phonetic classes out.

A. Maximum Likelihood Discriminant Projections

One common method of computing linear transformations to separate classes out is linear discriminant analysis (LDA) [16]. Given training data comprising a set of labeled feature vectors and denoting the ith feature vector for class j as $x_{j,i}$, the class-conditional sample means and covariances can be computed as

$$\mu_j = \frac{1}{N_j}\sum_i x_{j,i} \quad \Sigma_j = \frac{1}{N_j}\sum_i x_{j,i}x_{j,i}^T - \mu_j\mu_j^T$$

where N_j is the number of training vectors for class j and $N = \sum_j N_j$. The *average within-class covariance*, W and the *between-class covariance*, B, are computed as

$$W = \frac{1}{N}\sum_j N_j\Sigma_j \quad B = \frac{1}{N}\sum_j N_j\mu_j\mu_j^T - \bar{\mu}\bar{\mu}^T \qquad (3)$$

where $\bar{\mu}$ is the mean of the entire data. LDA finds a projection θ such that the average within-class variation in the projected space is minimized and the distance between the class means in the projected space is maximized. As the average-within-class covariance and between-class covariance in the projected space are given by $\theta W\theta^T$ and $\theta B\theta^T$, respectively, the LDA objective function is

$$J(\theta) = \frac{\left|\theta B\theta^T\right|}{\left|\theta W\theta^T\right|}. \qquad (4)$$

Even though the objective function (4) is nonlinear, there is a closed form solution given by the transposed eigenvectors corresponding to the p largest eigenvalues of the generalized eigenvalue problem: $Bx = \lambda Wx$.

LDA does not consider individual class covariances and may therefore generate suboptimal results. This shortcoming was recently addressed by modifying the LDA objective function [17], [18] as follows:

$$\prod_{j=1}^{J}\left(\frac{\left|\theta B\theta^T\right|}{\left|\theta\Sigma_j\theta^T\right|}\right)^{N_j} = \frac{\left|\theta B\theta^T\right|^N}{\prod_{j=1}^{J}\left|\theta\Sigma_j\theta^T\right|^{N_j}} \qquad (5)$$

leading to a heteroscedastic discriminant analysis (HDA) objective function. Fig. 1 shows the difference in the projections obtained from LDA and HDA for a 2-class case. Clearly, the HDA projection provides a much lower classification error than LDA. Taking the log and rearranging terms in (5), the HDA objective may be written as

$$H(\theta) = \sum_{j=1}^{J} -N_j\log\left|\theta\Sigma_j\theta^T\right| + N\log\left|\theta B\theta^T\right|. \qquad (6)$$

B. Bringing in the Assumption of Diagonal Covariances

The dimensions of the HDA projection can often be highly correlated, leading to a mismatch between HDA-projected features and the mixtures of diagonal covariance Gaussians that are most often used in acoustic models of ASR systems. Recent work [19], [20] has focused on finding a transformation that

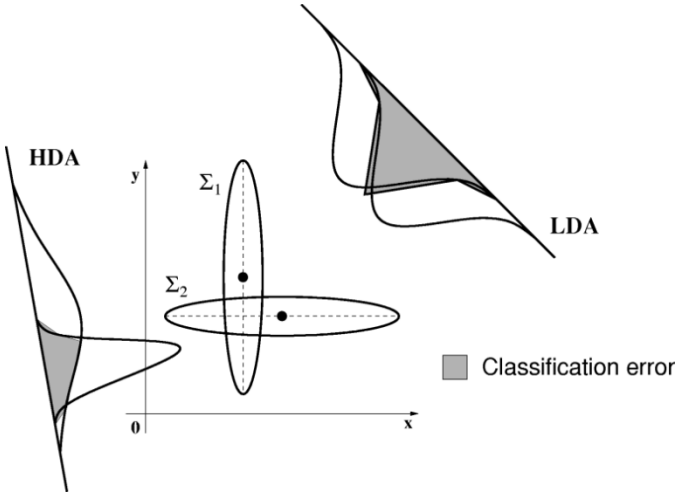

Fig. 1. Difference between LDA and HDA.

can be applied to correlated feature vectors, such that the transformed feature vector better satisfies the diagonal covariance assumption. The objective of these methods is to find a transformation, ψ, that can be composed with the projection, θ, such that the class-conditional covariances of the transformed feature vectors are as close to diagonal as possible. This is equivalent to minimizing the difference in the log-likelihoods of the data computed with full covariance and diagonal covariance models, i.e.,

$$\hat{\psi} = \arg\max_{\psi\in R^{p\times p}}\sum_{j=1}^{J}\frac{N_j}{2} \cdot\left(\log\left|\psi\hat{\Sigma}_j\psi^T\right| - \log\left|\operatorname{diag}\left(\psi\hat{\Sigma}_j\psi^T\right)\right|\right). \qquad (7)$$

Note that the first term of the HDA objective function (6) is essentially the log-likelihood of the projected feature vectors, using a single full covariance Gaussian per class and that the second term of the objective function is a measure of the scatter of the projected class means. Thus, maximizing the HDA objective function is equivalent to maximizing the log-likelihood of the data while simultaneously maximizing the separation between the class means. The HDA objective function is invariant to subsequent feature space transformations, hence the objective function (6) is the same for the composite transform $\psi\theta$ as for θ. We refer to the composite transform $\psi\theta$ as the *maximum likelihood discriminant (MLD) projection*. The HDA objective function (6) must be maximized using a generic optimization package such as the Fortran NAG library. The optimization for ψ (7) may be performed using an iterative scheme [21] or an optimization package.

C. Bringing in the Assumption of Canonical Feature Space and Speakers

The formulation presented in the previous sections assumes that the MLD projection performs dimensionality reduction at the first (speaker-independent) stage of processing. Features in a given class will therefore have variation due to both speaker-specific and intrinsic factors. For discrimination between phonetic classes, we are interested only in the intrinsic within-class

TABLE IV
PERFORMANCE FOR SYSTEMS USING TEMPORAL DERIVATIVES OR AN MLD PROJECTION TO INCLUDE TRAJECTORY INFORMATION IN THE FEATURE VECTOR

System	Features	E-VM1
VM1a	Mel Cepstra + deltas	40.5%
VM1b	Mel Cepstra + MLD proj.	39.1%
VM2a	PLP Cepstra + deltas	40.3%
VM2b	PLP Cepstra + MLD proj.	39.6%

variation. To achieve our objective, we can use speaker adaptation techniques [22] that improve the performance of speech recognition systems by "canonicalizing" the feature space (i.e., by eliminating as much of the speaker-specific variability as possible) prior to computing the MLD projection. Computing the MLD projection in the canonical space yields better relative improvements than in the original feature space.

D. Results

Table IV shows performance on the E-VM1 test set for systems using Mel cepstral features and either temporal derivatives (VM1a) or an MLD projection (VM1b), as well as systems using PLP features and temporal derivatives (VM2a) or an MLD projection (VM2b). The MLD projection provides a relative improvement of 1.7–3.5% over temporal derivatives.

We also tried using vocal tract length normalization (VTLN) [22] to reduce speaker-dependent, within-class variability prior to computing the MLD. More details on our VTLN implementation are provided in Section VI-A. Table V shows performance on the E-VM1 test set for systems using Mel cepstral features and either temporal derivatives (VM1c) or an MLD projection (VM1d), as well as systems using PLP features and temporal derivatives (VM2c) or an MLD projection (VM2d). Using Mel features, the MLD projection yields a 5.4% relative improvement over temporal derivatives in the canonical space versus a 3.5% relative improvement in the original feature space. In contrast, with PLP features the MLD projection degrades performance slightly, compared to temporal derivatives.[2]

V. HYPOTHESIS SEARCH

The most common decoding paradigm uses the maximum a posteriori (MAP) rule to guide the search. The MAP rule is

$$\underline{w}^* = \arg\max_{\underline{w}} p\left(\underline{w}|\underline{y}\right) = \arg\max_{\underline{w}} \frac{p\left(\underline{y}|\underline{w}\right) p\left(\underline{w}\right)}{p\left(\underline{y}\right)} \tag{8}$$

[2]This result on our PLP systems is not consistent with published results for Switchboard data, where an HLDA transform (which is very similar to the MLD transform) on VTLN PLP features produced relative improvements of 4–5% over temporal derivatives [2]. One possible explanation of the difference is that there is enough data from a given speaker in the Switchboard system that both mean and variance normalization are applied to the PLP features. In the voicemail system, only mean normalization is possible: variance normalization of short utterances was found to degrade performance. This has a bigger effect on the PLP systems compared to the MFCC systems because PLP cepstra are computed using cube-root compression, which is less compressive at high energies than the log compression used in the computation of Mel cepstra. Consequently, the PLP cepstra are more sensitive to the signal energy (in [2], presumably this is not a problem because the variance normalization compensates for varying signal energy) and this additional variability in the PLP cepstra degrades MLD performance on voicemail data.

TABLE V
PERFORMANCE FOR SYSTEMS USING TEMPORAL DERIVATIVES OR AN MLD PROJECTION TO INCLUDE TRAJECTORY INFORMATION IN THE FEATURE VECTOR. IN THESE TESTS THE FEATURES WERE "CANONICALIZED" USING VTLN

System	Features	E-VM1
VM1c	VTLN Mel Cepstra + deltas	38.9%
VM1d	VTLN Mel Cepstra + MLD proj.	36.8%
VM2c	VTLN PLP Cepstra + deltas	38.2%
VM2d	VTLN PLP Cepstra + MLD proj.	38.3%

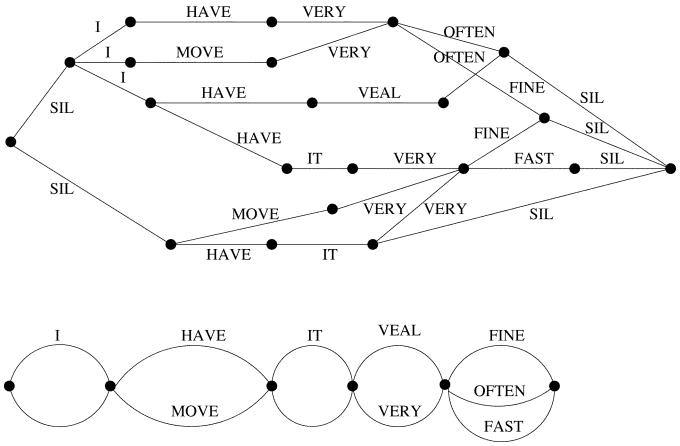

Fig. 2. Converting a word lattice to a confusion network.

where $\underline{w}$ is the sequence of decoded words and $\underline{y}$ is the sequence of observed feature vectors. An alternative search strategy [23] based on minimizing the average expected loss uses the following rule:

$$\underline{w}^* = \arg\min_{\underline{w}'} \sum_{\underline{w}} l\left(\underline{w}, \underline{w}'\right) p\left(\underline{w}|\underline{y}\right) \tag{9}$$

where $l(\underline{w}, \underline{w}')$ is the loss corresponding to $\underline{w}$ being the hypothesized sequence and $\underline{w}'$ being the reference word sequence. If $l(\underline{w}, \underline{w}')$ is taken to be a delta function, representing the sentence error rate, then (9) reduces to the MAP decoding rule (8). Hence, the MAP decoding rule is equivalent to minimizing the sentence error rate. Because most speech recognition applications try to minimize word error rate, it was suggested in [23] that $l(\underline{w}, \underline{w}')$ be replaced by the word error rate between the hypotheses $\underline{w}$ and $\underline{w}'$. In [23] this decoding rule was used to select a hypothesis from the N-best hypotheses produced by a MAP decoder.

In [24], this decoding rule was applied to a word lattice (produced by a MAP decoder) to obtain a "consensus hypothesis." The word lattice produced by the MAP decoder is first converted into a chain-like structure by merging different paths in the lattice. The criterion for merging two paths in the graph is related to the temporal overlap and phonetic similarity of the paths. The components of the resulting chain (a "confusion network") represent parallel sequences of words. This process is illustrated in in Fig. 2. After generation of the confusion network, the loss function evaluated over the complete sentence may be broken into a summation of K components, where each term in the

TABLE VI
WORD ERROR RATE WITH MAP AND CONSENSUS DECODING

System	Features	E-VM1 MAP	E-VM1 Consensus
VM1b	Mel Cepstra + MLD proj.	39.1%	38.0%
VM2b	PLP Cepstra + MLD proj.	39.6%	38.2%

TABLE VII
CONSENSUS WORD ERROR RATES FOR VTL NORMALIZED FEATURES

System	Features	E-VM1
VM1a[c]	Mel Cepstra + deltas	39.6%
VM1c[c]	VTLN Mel Cepstra + deltas	37.4%
VM1d[c]	VTLN Mel Cepstra + MLD proj.	35.0%
VM2a[c]	PLP Cepstra + deltas	39.1%
VM2c[c]	VTLN PLP Cepstra + deltas	37.1%
VM2d[c]	VTLN PLP Cepstra + MLD proj.	37.0%

summation corresponds to a component of the confusion network

$$l\left(\underline{w},\underline{w}'\right)=\sum_{k} l\left(w_k,w'_k\right). \tag{10}$$

The objective function in (9) may now be rewritten as

$$w_k^*=\arg\min_{w'_k}\sum_{w_k} l\left(w_k,w'_k\right)p\left(w_k|\underline{y}\right). \tag{11}$$

$l(w_k,w'_k)$ is a delta function for each component in the confusion network, thus approximating word error rate. Consequently

$$w_k^*=\arg\min_{w_k}\left[1-p\left(w_k|\underline{y}\right)\right] \tag{12}$$

and minimizing the loss is equivalent to picking the most probable word in each component. This is equivalent to applying the MAP rule in each component of the confusion network. The concatenation of these words is the consensus hypothesis. Further details are given in [24].

A. Results

Word error rates obtained by finding the consensus hypothesis rather than the MAP hypothesis are summarized for two systems in Table VI. There is a consistent 2.8–3.5% relative improvement in performance.

VI. ACOUSTIC MODEL ADAPTATION

Due to the widely varying nature of conversational speech, speaker-independent systems are often able to provide only a mediocre level of performance. One method that has emerged as an efficient way to improve system performance is adaptation. In adaptation, some samples of speech from a particular speaker (speaking over a specific channel, in a specific environment) are used to adapt the speech recognition features, models, or both so as to better match the speaker's speech. Both the speech from the test speaker and the associated transcription are necessary to perform adaptation. However, in the case of voicemail, the transcription of the speech data is not available. Therefore, a speaker-independent system is used to transcribe the data and the resulting (erroneous) transcriptions are used in the adaptation procedure. This is referred to as *unsupervised adaptation*. We experimented with two popular adaptation methods.

A. Vocal Tract Length Normalization (VTLN)

VTLN is based on the observation that a dominant source of difference between speakers is their pitch and formant ranges. Consequently, if the power spectrum for a speaker is scaled in frequency so that the formant frequencies for the speaker take on a target value (the target being the value of the formant frequency for a *canonical* speaker), then a significant source of inter-speaker variability in the acoustic feature vectors would be eliminated. The scaling of the frequency axis is assumed to be piecewise linear [22] and is parameterized by a slope a. This slope a is selected so as to maximize the likelihood of the speaker's data with respect to a canonical-speaker model, which models the cepstral feature vectors for each phonetic class for the canonical speaker. To normalize the feature vectors, the power spectrum is computed for a frame of speech from the speaker and the frequency axis is warped by one of a discrete set of values. The cepstra are then extracted from the frequency-warped spectra and the normalized[3] likelihood of the warped cepstra is computed with the model of the canonical speaker. This is done in an unsupervised manner using transcriptions produced by a speaker-independent system. The warp scale giving the highest likelihood is chosen and the speaker's data is warped accordingly. This procedure is carried out for all the speakers in the training data as well as the test data.

1) Results: Table VII summarizes experimental results obtained with VTLN. VTLN provides a relative improvement of 7.6–7.9% using temporal derivatives and 8.2–13.6% using an MLD projection.

B. Linear Transform Adaptation

While VTLN is highly effective, it does not completely eliminate speaker- and channel-dependent variability, nor does it produce a truly canonical feature space. It is therefore worthwhile to explore additional adaptation methods. We applied two methods based on linear transforms of the acoustic model in succession: 1) a feature-space realization of constrained maximum-likelihood linear regression [21] we call feature-space maximum-likelihood linear regression (FMLLR) and 2) standard MLLR [25].

[3]One subtlety associated with the selection of the warp scale involves the normalization of the likelihoods obtained for different warp scales. The VTLN warping can be thought of as a mapping from the original feature vectors, x_t, to a new set of vectors, y_t. It is possible to choose the mapping in such a way that the y_t have as little variance as possible and are as close as possible to the means of canonical speaker model. This would increase the likelihood without achieving the desired objective. Consequently, the likelihood computed with the canonical model has to be normalized by some measure of the covariance of the warped features. We do this by computing the covariance of the warped feature vectors for all of a speaker's data and normalize by the determinant of this covariance matrix.

TABLE VIII
CONSENSUS WORD ERROR RATES FOR SUCCESSIVE FMLLR AND MLLR ADAPTATION

System	Features	E-VM1
VM1d[c]	VTLN Mel Cepstra + MLD proj.	35.0%
VM1e[c]	VTLN Mel Cepstra + MLD proj. + FMLLR + MLLR	34.3%
VM2d[c]	VTLN PLP Cepstra + MLD proj.	37.0%
VM2e[c]	VTLN PLP Cepstra + MLD proj. + FMLLR + MLLR	35.5%

TABLE IX
WORD ERROR RATES WITH SYSTEM COMBINATION

	E-VM1	
System	MAP	Consensus
VM1b	39.1%	38.0%
VM1e	34.9%	34.3%
VM2b	39.6%	38.2%
VM2e	35.9%	35.5%
Rover	33.4%	32.7%

TABLE X
MAP ERROR RATES ON THE E-VM1 AND E-VM2 TEST SETS AS FUNCTION OF THE AMOUNT OF TRAINING DATA. THE SYSTEMS ARE PRESENTED IN ORDER OF DECREASING AMOUNT OF TRAINING DATA

Mel cepstra + MLD proj.

System	Training data	E-VM1	E-VM2
VM1b_l	T-VM1	30.2%	30.1%
VM1b	T-VM2	39.1%	38.0%
VM1b_m	T-VM4	41.9%	40.9%
VM1b_s	T-VM3	47.8%	47.0%

VTLN Mel cepstra + MLD proj. + FMLLR + MLLR

System	Training data	E-VM1	E-VM2
VM1e_l	T-VM1	27.9%	27.2%
VM1e	T-VM2	34.9%	33.7%
VM1e_m	T-VM4	38.1%	37.0%
VM1e_s	T-VM3	43.3%	42.7%

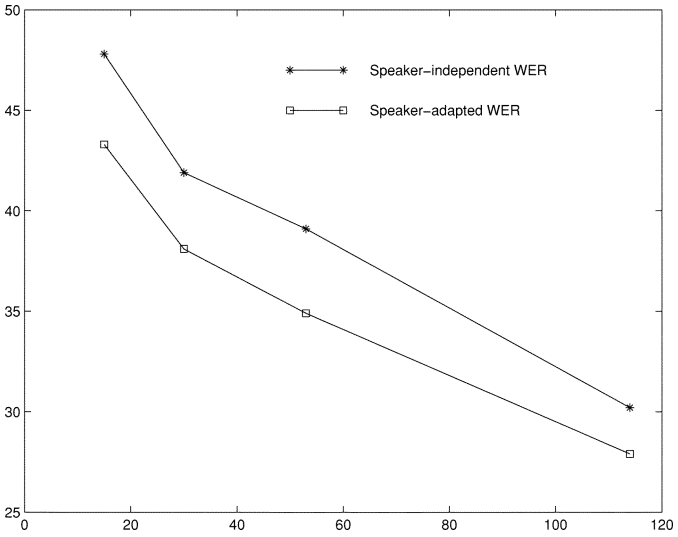

Fig. 3. MAP error rate on the E-VM1 test set as function of the amount of training data.

FMLLR learns an affine transformation of the adaptation data, x_t, of the form $y_t = A_f x_t + b_f$ that maximizes the likelihood of the transformed adaptation data, y_t, under the acoustic model. The FMLLR objective function has no closed form solution and must be solved using gradient descent or iterative methods [21].

Following FMLLR adaptation, the mismatch between the acoustic model and the transformed adaptation data can be further reduced by transforming the means of the acoustic model using MLLR [25]. MLLR learns one or more transformations, A_m^k, b_m^k of the model means based on the maximum-likelihood criterion. The number of transformations depends on the amount of available adaptation data. These affine transformations are shared by a subset of the Gaussians in the acoustic model, with sharing based on the distances between the Gaussians in the acoustic model (alternatively, prior phonetic knowledge may be used to structure the sharing). Further details are given in [25].

1) Results: Table VIII summarizes the results of applying these adaptation methods to voicemail data. Successive FMLLR and MLLR adaptation provides a relative improvement of 2–4% over the improvements provided by VTLN alone.

VII. SYSTEM COMBINATION

In the earlier sections, we presented results using a number of different systems. Though the average performance (WER) of most of these systems is similar, there are significant differences between their outputs. For instance, the hypotheses produced by VM1e and VM2e differ by 22%. In [26], a method was proposed in which the outputs of multiple ASR systems are treated as independent sources of knowledge and the outputs are combined to produce a composite output that had a lower WER than any of the individual outputs. The method is based on iteratively aligning multiple ASR outputs to form a word transition network (WTN) for an utterance. Subsequently, the composite WTN is searched using a voting or scoring module to select the best hypothesis.

A. Results

We applied the ROVER system combination technique to the outputs of the various systems that we experimented with. We

TABLE XI
SUMMARY OF THE CHARACTERISTICS OF ALL SYSTEMS DESCRIBED IN THIS PAPER

System	Features	Adaptation	# acoustic units	# Gaussians	Training Set
VM1a	Mel Cepstra + deltas		2063	68480	T-VM2
VM1b	Mel Cepstra + MLD proj.		2028	68676	T-VM2
VM1b_s	Mel Cepstra + MLD proj.		1808	35347	T-VM3
VM1b_m	Mel Cepstra + MLD proj.		2009	54351	T-VM4
VM1b_l	Mel Cepstra + MLD proj.		2778	259876	T-VM1
VM1c	Mel Cepstra + deltas	VTLN	2063	68480	T-VM2
VM1d	Mel Cepstra + MLD proj.	VTLN	2104	70426	T-VM2
VM1e	Mel Cepstra + MLD proj.	VTLN + FMLLR + MLLR	2104	70426	T-VM2
VM1e_s	Mel Cepstra + MLD proj.	VTLN + FMLLR + MLLR	1877	35883	T-VM3
VM1e_m	Mel Cepstra + MLD proj.	VTLN + FMLLR + MLLR	2044	54381	T-VM4
VM1e_l	Mel Cepstra + MLD proj.	VTLN + FMLLR + MLLR	3112	265686	T-VM1
VM2a	PLP Cepstra + deltas		2051	68300	T-VM2
VM2b	PLP Cepstra + MLD proj.		2022	68944	T-VM2
VM2c	PLP Cepstra + deltas	VTLN	2051	68300	T-VM2
VM2d	PLP Cepstra + MLD proj.	VTLN	2043	69140	T-VM2
VM2e	PLP Cepstra + MLD proj.	VTLN + FMLLR + MLLR	2043	69140	T-VM2

tested both combinations of MAP decoding outputs and combinations of consensus decoding outputs. Table IX summarizes the results. We observe a consistent relative improvement of 4.3–4.6% from combining the outputs of four systems.

VIII. EFFECT OF TRAINING DATA

All experimental results reported in the previous sections were trained on the T-VM2 database. To characterize the effect of the amount of training data on system performance, we also experimented with systems trained on the complete Voicemail corpus (T-VM1, comprising 114 h of speech), the T-VM3 subset (comprising 15 h of speech) and the T-VM4 subset (comprising 30 h of speech). Increasing the amount of training data improves the recognition system in a number of ways:

- the resolution of the acoustic model improves because the numbers of acoustic units and Gaussian mixture components can be increased;
- the language model becomes more accurate with increased in-domain training data;
- out-of-vocabulary rates on test data decrease because the lexicon is designed to cover the most frequent 99% of the training words.

The results are summarized in Table X and plotted in Fig. 3. Both the speaker-independent and speaker-adapted error rates appear to decrease almost linearly with increasing amounts of training data, except for the smallest training set where the improvement appears to be superlinear. This result is contrary to the conventional wisdom that word error rate decreases in proportion to the log of the amount of training data. We surmise that this relationship holds only for much larger amounts of training data.

IX. CONCLUSION

In this paper, we report on the evolution of the word error rate (WER) on a large vocabulary telephone speech recognition task, as typified by voicemail. A number of algorithms were developed and evaluated in the context of this task that contributed significantly to reducing the WER; these algorithms span the areas of lexicon design, feature extraction, search, adaptation of acoustic models and hypothesis combination and were instrumental in reducing the word error rate on voicemail data to around 27.9%. In the area of lexicon design we presented a data-driven method for adding compound words to the lexicon that yields up to a 3.6% relative improvement in performance. In the area of feature extraction we presented a linear projection technique (MLD) that yields up to a 5.4% relative improvement in performance. We also showed that the MLD is better applied to features that have already been canonicalized to eliminate inter-speaker variation. In the area of search we showed that the use of a consensus hypothesis algorithm yields up to a 3.5% relative improvement in performance. In the area of adaptation we showed that VTLN adaptation provides up to 4.1% relative improvement in performance and that adaptation by successive linear transforms (FMLLR + MLLR) provides up to a 4% relative improvement in performance. In the area of system combination we showed that using ROVER to combine the outputs of four systems yields up to a 4.6% relative improvement in performance. We also demonstrated roughly linear improvements

in word error rate with increasing amounts of training data and showed that adaptation provides a relative improvement of 9.4% for a system trained on 15 h of data, but only an improvement of 7.6% for a system trained on 114 h of data. This result is consistent with the conventional wisdom that performance improvements due to adaptation decline as a system is trained on more data.

The MLD, consensus decoding, adaptation and system combination techniques are general methods that have been shown to improve system performance on a number of speech recognition tasks. Each of these methods can be explained in terms of basic principles in pattern recognition. The MLD projection is a form of discriminative training, consensus decoding bases the hypothesis search directly on the system performance measure (word error rate), adaptation reduces variability that is not related to class discrimination and system combination is simply a form of voting. The improvement we observed due to compound word modeling may be more task-specific, however, because the source of this improvement appears to be better modeling of "stock phrases" in voicemail data. It is possible that compound word modeling could yield smaller improvements on more heterogeneous data.

APPENDIX

The systems reported on in this paper are summarized in Table XI by features, adaptation, number of acoustic units, number of Gaussian mixtures, and training data.

ACKNOWLEDGMENT

The authors are grateful to B. Ramabhadran, S. Chen and G. Zweig of the Human Language Technologies Department for their assistance.

REFERENCES

[1] J. Godfrey, E. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. ICASSP*, 1992.

[2] P. C. Woodland, G. Evermann, M. Gales, T. Hain, A. Liu, G. Moore, D. Povey, and L. Wang, "CU-HTK April 2002 switchboard system," in *Proc. NIST RT-02 Workshop*, 2002.

[3] J. Huang, G. Zweig, and M. Padmanabhan, "Information extraction from voicemail," in *Proc. Meeting Association for Computational Linguistics*, 2001.

[4] M. Bacchiani, J. Hirschberg, A. Rosenberg, S. Whittaker, D. Hindle, P. Isenhour, M. Jones, L. Stark, and G. Zamchick, "SCANMail: Audio navigation in the voicemail domain," in *Proc. Human Language Technology Conf.*, 2001.

[5] J. Hirschberg, M. Bacchiani, D. Hindle, P. Isenhour, A. Rosenberg, L. Stark, L. Stead, S. Whittaker, and G. Zamchick, "SCANMail: Browsing and searching speech data by content," in *Proc. EUROSPEECH*, 2001.

[6] M. Padmanabhan, G. Ramaswamy, B. Ramabhadran, P. S. Gopalakrishnan, and C. Dunn, "Issues involved in voicemail data collection," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[7] K. Koumpis and S. Renals, "Transcription and summarization of voicemail speech," in *Proc. ICSLP*, 2000.

[8] M. Bacchiani, "Automatic transcription of voicemail at AT&T," in *Proc. ICASSP*, 2001.

[9] R. Córdoba, P. C. Woodland, and M. J. F. Gales, "Improved cross-task recognition using MMIE training," in *Proc. ICASSP*, 2002.

[10] L. R. Bahl, P. V. de Souza, P. S. Gopalakrishnan, D. Nahamoo, and M. A. Picheny, "Robust methods for using context-dependent features and models in a continuous speech recognizer," in *Proc. ICASSP*, 1994.

[11] P. S. Gopalakrishnan, L. R. Bahl, and R. L. Mercer, "A tree search strategy for large vocabulary continuous speech recognition," in *Proc. ICASSP*, 1995.

[12] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 357–366, Aug. 1980.

[13] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *J. Acoust. Soc. Amer.*, vol. 87, no. 4, pp. 1738–1752, Apr. 1990.

[14] G. Saon and M. Padmanabhan, "Data-driven approach to designing compound words for continuous speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 9, pp. 327–332, May 2001.

[15] M. Finke and A. Waibel, "Speaking mode dependent pronunciation modeling in large vocabulary conversational speech recognition," in *Proc. EUROSPEECH*, 1997.

[16] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic, 1990.

[17] N. Kumar and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMS for improved speech recognition," *Speech Commun.*, vol. 26, pp. 283–297, 1998.

[18] G. Saon, M. Padmanabhan, R. A. Gopinath, and S. Chen, "Maximum likelihood discriminant feature spaces," in *Proc. ICASSP*, 2000.

[19] R. A. Gopinath, "Maximum likelihood modeling with Gaussian distributions for classification," in ICASSP, 1998.

[20] M. J. F. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Trans. Speech Audio Processing*, vol. 7, pp. 272–281, May 1999.

[21] ——, "Maximum likelihood linear transforms for HMM-based speech recognition," *Comput. Speech Lang.*, vol. 12, 1998.

[22] S. Wegman, D. McAllaster, J. Orloff, and B. Peskin, "Speaker normalization on conversational telephone speech," in *Proc. ICASSP*, 1996.

[23] A. Stolcke, Y. Konig, and M. Weintraub, "Explicit word error minimization in N-best list rescoring," in *Proc. EUROSPEECH*, 1997.

[24] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus among words: Lattice-based word error minimization," in *Proc. EUROSPEECH*, 1999.

[25] C. J. Legetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Comput. Speech Lang.*, vol. 9, 1995.

[26] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 1997.

Mukund Padmanabhan (S89–M'89–SM'99) received the B.Tech degree in electronics and communication engineering from the Indian Institute of Technology, Kharagpur, and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Los Angeles.

His interests range over a large number of areas spanning communications, signal processing, analog integrated circuits, speech recognition, and statistical financial modeling. He worked in the area of speech recognition at the IBM T. J. Watson Research Center, Yorktown Heights, NY, from 1992 to 2001, where he managed the Telephony Speech Algorithms Group. Currently, he works for Renaissance Technologies in the area of financial modeling. He is an editor for the *Journal on Applied Signal Processing*. He is the co-author of a book on signal processing and circuits entitled *Feedback-Based Orthogonal Digital Filters: Theory, Applications, and Implementation*.

Dr. Padmanabhan was a recipient of one of the Best Paper Awards (for a paper in the IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING) from the IEEE Signal Processing Society in 2001. He is a member of the IEEE SPS Speech Technical Commitee.

George Saon received the M.Sc. and Ph.D. degrees in computer science from the Université Henri Poincaré, Nancy, France, in 1994 and 1997, respectively.

From 1994 to 1998, he worked on stochastic modeling for offline handwriting recognition at the Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA). He is currently a Research Staff Member with the Human Language Technologies Department, IBM T. J. Watson Research Center, Yorktown Heights, NY, where he is conducting research on large vocabulary conversational speech recognition over the telephone. His research interests are in pattern recognition and stochastic modeling.

Jing Huang received the B.S. degree in applied mathematics from Tsinghua University, Beijing, China, in 1990 and the Ph.D. degree in computer science from Cornell University, Ithaca, NY, in 1998, working on computer vision and image retrieval systems.

She is currently a Research Staff Member with the Human Language Technologies Department at the IBM T. J. Watson Research Center, Yorktown Heights, NY. Her research interests now include speech recognition, natural language processing, and information management systems.

Brian Kingsbury received the B.S. degree in electrical engineering from Michigan State University, East Lansing, in 1989 and the Ph.D. degree in computer science from the University of California, Berkeley, in 1998.

He is currently a Research Staff Member with the Human Language Technologies Department at the IBM T. J. Watson Research Center, Yorktown Heights, NY. His research interests include acoustic modeling and robustness in large vocabulary conversational speech recognition systems.

Lidia Mangu received the B.S. degree from the University of Bucharest, Romania, in 1992 and the M.S. and Ph.D. degrees in computer science from the Johns Hopkins University, Baltimore, MD, in 1997 and 2000, respectively.

She is currently a Research Staff Member in the Human Language Technologies Department at the IBM T. J. Watson Research Center, Yorktown Heights, NY. Her past and present research interests span language modeling and decoding algorithms for speech recognition and computational linguistics.