RC22172 (W0109-014) September 14, 2001
Computer Science

IBM Research Report

Automatic Speech Recognition Performance on a Voicemail Transcription Task

Mukund Padmanabhan, George A. Saon, Jing Huang, Brian Kingsbury, Lidia L. Mangu
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

LIMITED DISTRIBUTION NOTICE: This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. It has been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies of the article (e.g. , payment of royalties). Copies may be requested from IBM T. J. Watson Research Center , P. O. Box 218, Yorktown Heights, NY 10598 USA (email: reports@us.ibm.com). Some reports are available on the internet at http://domino.watson.ibm.com/library/CyberDig.nsf/home .

Automatic Speech Recognition Performance on A Voicemail Transcription Task

M. Padmanabhan, G. Saon, J. Huang, B. Kingsbury, L. Mangu
IBM T. J. Watson Research Center
P. O. Box 218, Yorktown Heights, NY 10598
e-mail: (mukund,gsaon,yellow,bedk,mangu)@us.ibm.com

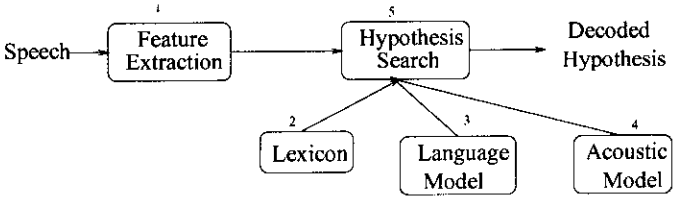

Fig. 1. Speech Recognition System Block Diagram

***Abstract*—In this paper we report on the performance of ASR systems on a fairly difficult problem - that of transcribing voicemail. Voicemail represents spontaneous telephone speech recorded over a variety of channels - consequently, it represents a testbed that contains all the most challenging problems in speech recognition. In the course of working on this problem, several algorithms were developed that focus on different building blocks of a typical ASR system, such as lexicon design, feature extraction, hypothesis search and speaker adaptation. The goal of this paper is to benchmark the improvements provided by these algorithms, as well as other standard techniques, on a voicemail test set. Though 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.**

I. Introduction

In the last few years, several advances have been made in improving the error rate of continuous-speech-recognition systems. For instance, the word-error rates on read speech (as in the Wall Street Journal database) hover in the neighborhood of 7-8 %. However, the performance on sources of spontaneous speech is still relatively poor. For instance, word-error rates on the Switchboard corpus [1] and Voicemail corpora are around 30%. This is all the more unfortunate because spontaneous speech is much more characteristic of real-world speech. Consequently, there is still a great deal of improvement that has to be made to make speech recognition systems practical and useable in real life. 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.

There are several steps involved in speech recognition systems today, as shown in Fig. 1. In Block 1, multi-dimensional features are extracted from the sampled speech signal; in Block 5, a search procedure hypothesizes a word sequence that has the maximum probablility given the observed feature vectors. This step is driven by several components (i) a lexicon (Block 2) that defines the possible words that can be hypothesized in the search and where each word is represented as a linear sequence of fundamental phonetic units, (ii) a language model (Block 3) that models the linguistic structure – this model incorporates purely linguistic knowledge and does not contain any knowledge about the relationship between the feature vectors and the words, (iii) an acoustic model (Block 4) that models the relationship between the feature vectors and the fundamental phonetic units – the fundamental phonetic units are modeled using HMM's and the output distributions of the HMM's 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 in Fig. 1, 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, and the various systems that we experimented with. In Section III we address the problem of lexicon design (Block 2 of Fig. 1), and describe a data driven method, to find compound words to augment the lexicon. In Section IV, we describe experiments related to the acoustic model (Block 1 of Fig. 1), 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 graph (Block 5 of Fig. 1). Finally, in Section VI, we describe the performance improvements obtained by adapting the acoustic models to the specific test speaker.

[0] We would like to acknowledge the support of DARPA under Grant MDA972-97-C-0012 for funding part of this work.

II. Background

A. Training/Test data

The Voicemail training database comprises 12645 messages (114 hours of speech), which corresponds to approximately 1.167M words of text. We will refer to this training database as T-VM1. In order to facilitate a quicker turaround of experiments, and to provide results on publicly available data sets, and toe examine the effect of training data on system performance, we defined several subsets of this database

- T-VM2 comprises of 6501 messages (53 hours of speech)
- T-VM3 comprises of 1801 messages (15 hours of speech) - available as Voicemail Corpus Part I from LDC
- T-VM3b comprises of 2048 messages (15 hours) - soon to be made available as Voicemail Corpus Part II from LDC
- T-VM4 - comprises of all the messages in T-VM3 and T-VM3b

The size of the testing vocabulary is 19k words for systems trained on T-VM1, 11.5k for systems trained on T-VM2, and 8.8k words for systems trained on T-VM4 and 6.5k words for systems trained on T-VM3. The test set comprises of 105 messages (of these, 42 are currently available as part of the LDC Voicemail Corpus Part I), and we expect to make an additional 50 messages from this set available as part of the Voicemail Corpus Part II. The language model is a trigram. We will refer to the test set with 105 messages (52 minutes of speech) as E-VM1. We also define a subset of this test set that includes 92 messages (35 minutes of speech) and refer to this as E-VM2.

B. System Description

The speech recognition system uses a phonetic representation of the words in the vocabulary. Each phone is modelled with a 3-state left-to-right HMM. Further, we identify the variants of each state that are acoustically dissimilar by asking questions about the phonetic context in which the state occurs. The questions are arranged hierarchically in the form of a decision tree, and its leaves correspond to the basic acoustic units that we model. A feature vector is extracted every 10 ms, and we model the probability density function (pdf) of the feature vector for each leaf of the decision tree with a mixture of gaussians. However, rather than use the probability density provided by this model, we use a "rank" based system, where all the leaves of the decision tree are first ranked on the basis of their probablity densities for a particular feature vector. Subsequently, when we need to compute the probability of an observation given a leaf, we look up the rank of that leaf, and convert that to a probability using a table (for example, leaves with rank of 1 invariably will have a higher probability than leaves with a much poorer rank). For futher details, see [4]. The hypothesis search that is used to find candidate word sequences that have a reasonable probability is a modified version of the stack search, that we refer to as an envelope search. For further details, see [5].

System	FSP	#L	#G	Trg
VM1a	Mel Ceps	2063	68480	T-VM2
VM1b	Mel Ceps + MLD	2028	68676	T-VM2
VM1b_s	Mel Ceps + MLD	1808	35347	T-VM3
VM1b_m	Mel Ceps + MLD	2009	54351	T-VM4
VM1b_l	Mel Ceps + MLD	2778	259876	T-VM1
VM1c	Mel Ceps + VTL	2063	68480	T-VM2
VM1d	Mel Ceps + VTL + MLD	2104	70426	T-VM2
VM1e	Mel Ceps + VTL + MLD	2104	70426	T-VM2
VM1e_s	Mel Ceps + VTL + MLD	1877	35883	T-VM3
VM1e_m	Mel Ceps + VTL + MLD	2044	54381	T-VM4
VM1e_l	Mel Ceps + VTL + MLD + FMLLR + MLLR	3112	265686	T-VM1
VM2a	PLP Ceps	2051	68300	T-VM2
VM2b	PLP Ceps + MLD	2022	68944	T-VM2
VM2c	PLP Ceps + VTL	2051	68300	T-VM2
VM2d	PLP Ceps + VTL + MLD	2043	69140	T-VM2
VM2e	PLP Ceps + VTL + MLD + FMLLR + MLLR	2043	69140	T-VM2

TABLE I
Systems description

The baseline feature vector is the Mel cepstrum [8] augmented with its 1st and 2nd temporal derivatives (which we refer to as deltas). We will refer to this as the Mel cepstral feature space. We also present experiments that used the PLP cepstra [9] instead of the Mel cepstra. Both the Mel and PLP cepstra were mean-normalized on a per-sentence basis. Some of the systems that we experimented with spliced together 9 frames of cepstra (the cepstra at the current frame and 4 frames before and after the current frame) and projected it down to a lower dimension by means of a linear transform. We will refer to this feature space as the projected feature space.

During the course of running these experiments, we built a number of "baseline" acoustic models The improvements accruing from specific algorithms are benchmarked on these baseline systems. We summarize the models that we worked with in Table I. The column FSP indicates the type of feature space (the dimensionality of the space is 39 for all systems), $\#L$ indicates the number of leaves in the decision tree, $\#G$ indicates the number of gaussians, and Trg indicates the training data that was used to build the system.

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 contains all words that could be uttered by the speaker, and the words are dissimilar and can be easily disambiguated on the basis of the language and acoustic model, this would lead to very good accuracy in speech recognition. This is unfortunately not the case, however, it is generally the case that decoding errors are more common in shorter words [6]. One possible method to capitalize on this observation is to combine groups of words that co-occur, into compound words. As these compound words would now have relatively long baseforms, they could lead to fewer overall errors. Another observation in connection with spontaneous speech is that crossword co-articulation is very common because of the casual nature of the speech and the fast speaking rate. For instance, the phrase 'going to take' could be pronounced as 'gontake = G AO N T AE KD'. Another advantage associated with having compound words is that these co-articulation effects can be modeled by adding the appropriate baseform for the co-articulated compound word.

A. Measures to select compound words

Though the motivation for adding compound words to the vocabulary is clear, adding more tokens or pronunciation variants to the acoustic vocabulary and/or the language model could also increase the confusability between words. Hence, the candidate pairs for compound words have to be chosen carefully in order to avoid this increase. Intuitively, such a pair has to meet several requirements:

- The pair of words has to occur frequently in the training corpus.
- The words within the pair have to occur frequently together and more rarely in the pair context of other words.
- The words should ideally present coarticulation effects at the juncture i.e., their continuous pronunciation should be different than when they are uttered in isolation.

In [7], the mutual information between a pair of words was used to select compound words. Though this measure picks words that occur frequently together, it does not guarantee 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 [6] – to rank word pairs and select compound words. 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 as follows:

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

Sys.	Nb.	Examples	E-VM1
VM1a	0		42.0%
VM1a	70	AREA-CODE, GIVE-ME, A-CALL, E-MAIL, TAKE-CARE GIVE-ME-A-CALL, LET-ME-KNOW, AS-SOON-AS, THANK-YOU-VERY-MUCH TALK-TO-YOU-LATER-BYE, THANKS-A-LOT, PLEASE-GIVE-ME-A-CALL	40.5%

TABLE II
WORD ERROR RATE FOR LEXICON WITH COMPOUND WORDS

The geometrical average of the direct and the reverse bigram is given by:

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

$0 \leq LM(w_i, w_j) \leq 1$ 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 bigrams are high for (w_i, w_j) or otherwise stated, the probabilities that w_i is followed by w_j and w_j is preceded by w_i are high which makes the pair a good candidate for a compound word. In our implementation we selected all pairs of words for which this measure is greater than a fixed threshold and for which the raw count of the word pair was greater than another threshold.

B. Results

We report results here with the compound-words selected according to (2). We started with a lexicon with no compund words and applied the measure iteratively to the corpus resulting in increasing number of compound words per iteration. Table II summarizes the total number of new compound words, examples of such words, and the word error rate. In summary, it may be seen that adding compound words based on the LM measure results in a 1.5% absolute (3.6% relative) improvement in the word error rate. This vocabulary (with compound words) and the associated trigram LM will be used in all Voicemail related experiments in subsequent sections.

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). Quite often the procedure for extracting this feature vector is motivated by the workings of the human auditory system. As the human auditory system simulates a log-spaced filter-bank, with the sensitivity to the energy in each channel following a logarithmic relationship, most feature extraction schemes (examples are Mel cepstra [8] and PLP cepstra [9]) are based on mimicking these steps.

The d-dimensional feature vectors extracted in this manner every 10ms contain information about the local spectral characteristics of the speech signal, however, they do not contain any information related to the trajectory of the

spectral characteristics over time. In order to incorporate this information, one possiblity is to compute the temporal derivative of each of the d dimensions, and concatenate these values to the d-dimensional cepstra. In addition, it is also customary to compute the second derivative and append those values to the cepstra. Consequently, the final feature extracted every 10ms has a dimensionality of $3d$.

Another possibility for incorporating temporal trajectory information is by means of a linear discriminant transformation. In this approach, the d-dimensional cepstra from several adjacent frames (typically 9) are concatenated together to form a $9d$-dimensional feature vector. The dimensionality of the feature vector is then reduced by computing a linear projection that finds the directions that maximally separate the phonetic classes out. In this section, we describe a process of computing a linear transformation on the cepstra 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) [10]. Let us assume that we are provided with training data comprising a set of feature vectors for each phonetic class. Denoting the i^{th} feature vector for class j as $x_{j,i}$, the mean and covariance of the data for each class 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 denotes the number of training vectors for class j, and $N = \sum_j N_j$. A measure of the intra-class variation of the feature vectors, referred to as the *average within-class covariance*, and the distance between the class means, quantified by a matrix called the *between-class covariance* matrix, are given by

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

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

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

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

In the LDA objective function, the average-within-class-covariance was minimized without considering the individual class covariances. Recently [11], [12], this shortcoming was addressed by modifying the LDA objective function (4) as follows:

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

leading to a *HDA* objective function. Fig 2 shows the difference in the projections obtained from LDA and HDA for the 2-class case. Clearly, the HDA projection provides a much lower classification error than LDA. Taking log

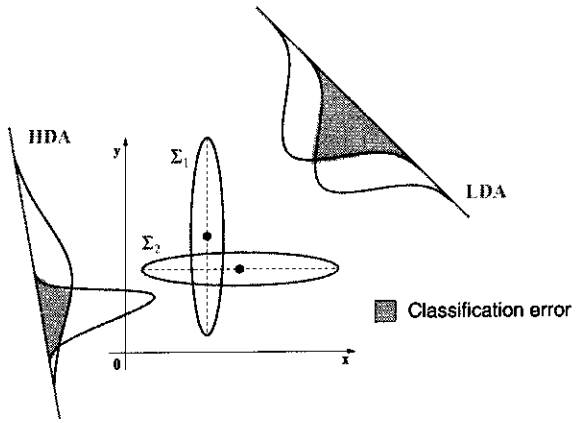

Fig. 2. Difference between LDA and HDA.

and rearranging terms in (5), the HDA objective may be written as:

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

Another interpretation of the objective function in (6) arises by noting that the first term (the summation) essentially equals the log-likelihood of the projected feature vectors, assuming a single full covariance gaussian model for each class, with mean and covariance given by $\hat{\mu}_j = \theta\mu_j$ and $\hat{\Sigma}_j = \theta\Sigma_j\theta^T$, $1 \le j \le J$. The second term, $N\log|\theta B\theta^T|$ represents the distances between the class means, $\hat{\mu}_j$, in the projected feature space. Consequently, the objective function (6) is really trying to maximize the log-likelihood of the data while at the same time maximizing the separation between the class means – consequently we refer to this as the *Maximum Likelihood Discrimimant (MLD) Projection.*

B. Bringing in the assumption of diagonal covariances

One of the characteristics of the projection described above is that the dimensions of the projected feature vectors often tend to be highly correlated. However, due to computational considerations, most ASR systems model the pdf of the projected feature vectors with mixtures of

diagonal gaussians, which in effect assumes that the dimensions are independent. The inaccuracies associated with this assumption often nullify any benefit associated with designing the transformation in the first place. Consequently, recent work [13], [14] has focused on finding a transformation that can be applied to the correlated feature vectors, such that the transformed feature vector more or less 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 covariance of the transformed feature vectors is as close to diagonal as possible. This is equivalent to minimizing the difference in the log-likelihood of the data, computed with full covariance and diagonal covariance models, i.e.,

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

Note that the objective function (6) is invariant to subsequent feature space transformations, hence the objective function (6) is the same for the composite transform $\psi\theta$ as for θ. Consequently, the process of finding the MLD projection first involves finding θ to maximize (6) and subsequently finding a ψ to maximize (7). Both objective functions have no closed-form solutions; consequently, optimization packages such as the Fortran NAG library have to be used to find the solution.

C. Bringing in the assumption of canonical feature space and speakers

The formulation presented in the previous sections assumed that the projection is computed to perform dimensionality reduction at the first (speaker-independent) stage of the processing. Now, the main objective of the projections was to minimize the variation of the projected feature within a particular class, while maximizing the distance between the projected means of the different classes. As the training data for a speaker independent system is comprised of speech from a number of different speakers, the variation of the projected features within a particular class has an inherent component as well as an inter-speaker component. For the purposes of discriminating between phonetic classes, we are really interested only in focusing on the inherent variation, rather than the inter-speaker variation. In order to achieve our objective, we can take the help of speaker adaptation techniques [18] that are focused on improving the performance of speech recognition systems by "canonicalizing" the feature space i.e. by eliminating as much of the inter-speaker variability as possible. This is in effect equivalent to first "canonicalizing" the feature space with some speaker normalizing scheme and then computing a discriminant transform that separates the phonetic classes out in the canonicalized space. We present experimental results in the next section that show that computing the MLD projection in the canonical space yields better relative improvements than the original feature space.

System	Feature space	E-VM1
VM1a	Mel Cepstra	40.5%
VM1b	Mel Cepstra + MLD Proj	39.1%
VM2a	PLP Cepstra	40.3%
VM2b	PLP Cepstra + MLD Proj	39.6%

TABLE III
WORD ERROR RATE FOR CEPSTRA AND MLD FEATURES

System	Feature space	E-VM1
VM1c	Mel Cepstra + VTLN	38.9%
VM1d	Mel Cepstra + VTLN + MLD Proj	36.8%
VM2c	PLP Cepstra + VTLN	38.2%
VM2d	PLP Cepstra + VTLN + MLD Proj	38.3%

TABLE IV
WORD ERROR RATE FOR "CANONICALIZED" CEPSTRA AND MLD FEATURES

D. Results

The word error rates obtained on the voicemail test set (E-VM1) for the cepstral and projected feature spaces are shown in Table III. The models and vocabulary were described in Section II-B and Section III-B The MLD projection was computed on the baseline mean-normalized Mel and PLP cepstra, and can be seen to provide a relative improvement of 3.5%-1.7% over the baseline cepstral space.

We also experimented with first "canonicalizing" the feature space and then computing the MLD in the canonical space. For the purpose of these experiments, we chose VTLN [18] as the canonicalization scheme. Details of VTLN are given in Section VI-A however, for the purpose of these experiments, it is sufficient to just treat VTLN as some scheme for removing the inter-speaker variability in the feature vectors. The results are presented in Table IV and show that the MLD projection yields a relative improvement of 5.4% in performance for the Mel cepstra. Interestingly enough, the MLD projection consistently yields less improvement for the PLP cepstral space compared to the Mel cepstral space.

V. HYPOTHESIS SEARCH

The most commonly used decoding paradigm for speech recognition is the maximum aposteriori (MAP) rule which is used to guide the hypothesis search.

$$\underline{w}^* = argmax_{\underline{w}} p(\underline{w}/\underline{y}) = argmax_{\underline{w}} p(\underline{y}/\underline{w}) p(\underline{w}) / p(\underline{y}) \quad (8)$$

where $\underline{w}$ represents the sequence of decoded words and $\underline{y} = y_1^T$ denotes the observed feature vectors corresponding to the sentence. In [15], an alternative scoring procedure

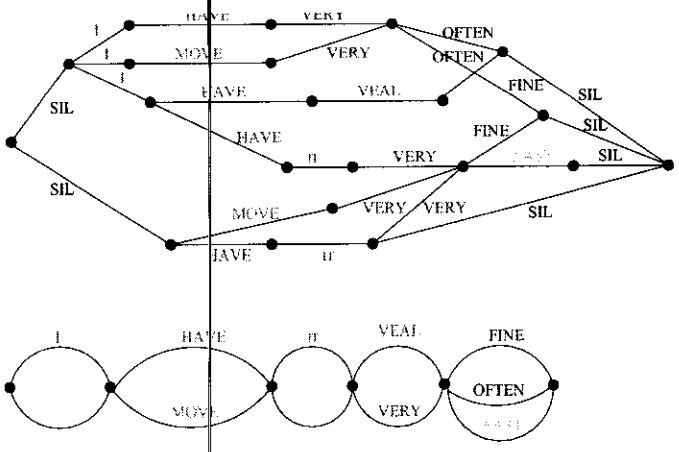

Fig. 3. Converting a word lattice to a confusion network

was proposed to guide the hypothesis search. If $l(\underline{w}, \underline{w}')$ represents the loss, corresponding to $\underline{w}$ being the hypothesized sequence and $\underline{w}'$ being the reference word sequence, then the objective of the decoding procedure should be to minimize the average expected loss. This may be written

$$\underline{w}^* = argmin_{\underline{w}'} \sum_{\underline{w}} l(\underline{w}, \underline{w}')p(\underline{w}/\underline{y}) \quad (9)$$

If $l(\underline{w}, \underline{w}')$ is taken to be a delta function, representing the sentence error rate, then (9) can be seen to reduce to the MAP decoding rule. Hence, the MAP decoding rule is equivalent to minimizing the sentence error rate. However, as the quantity of interest in most speech recognition applications is the word error rate, it was suggested in [15] that $l(\underline{w}, \underline{w}')$ be replaced by the word error rate between the hypotheses $\underline{w}$ and $\underline{w}'$. In [15] this decoding rule was implemented to select a hypothesis from the N-best hypotheses produced by a MAP decoder. In [16] this decoding rule was applied to a word lattice (that was produced by a MAP decoder) to obtain a "consensus hypothesis" as follows: The word lattice (graph) produced by a MAP decoder is first converted into a chain like structure by merging different paths in the graph. This is shown in Fig 3.

The components of the chain represent parallel sequences of words. The criterion for merging two paths in the graph is related to the time overlap between the paths and the phonetic similarity between the word sequences in the two paths. The loss function evaluted over the complete sentence may now be broken into K components, where each term in the summation corresponds to a component of the chain.

$$l(\underline{w}, \underline{w}') = \sum_k l(w_k, w_k') \quad (10)$$

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

$$w_k^* = argmin_{w_k'} \sum_{w_k} l(w_k, w_k')p(w_k/\underline{y}) \quad (11)$$

For $l(w_k, w_k')$ to emulate the word error rate, as each component of the chain contains only words in parallel, it has to be defined as a delta function. Consequently we have

$$w_k^* = argmin_{w_k}[1 - p(w_k/\underline{y})] \quad (12)$$

and minimizing the loss is equivalent to picking the most probable word in each component (which is equivalent to applying the MAP rule in each component of the chain). The concatenation of these words represents the consensus hypothesis. Further details are given in [16].

System	Feature space	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 V
WORD ERROR RATE WITH MAP AND CONSENSUS HYPOTHESIS

A. Results

The word error rates obtained by finding the consensus hypothesis rather than the MAP hypothesis are summarized in Table V. It may be seen that there is a consistent 2.8%-3.5% relative improvement in performance.

VI. Acoustic Model - Adaptation

Due to the widely varying nature of speech, and the difficulty of the speech recognition task, speaker-independent systems are only able to provide a mediocre level of performance on this task. One of the methods that has emerged as an efficient way to improve system performance in such cases is speaker adaptation. In speaker adaptation, some samples of speech from a particular speaker are used to adapt the speech recognition models so as to better match the test speakers speech. In order to adapt the models, both the speech from the test speaker as well as the associated transcription are necessary. However, in the case of voicemail, as the transcription of the speech data is not available, 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 that are summarized in the next two sections.

A. Vocal Tract Length Normalization (VTLN)

This technique is based on the observation that the models in a speaker-independent ASR system are forced to model the pdf of the feature vectors for different phonetic classes and different speakers. If the inter-speaker variation in these feature vectors is large, it could dominate the overall variance of the model. However, the goal of speech recognition systems is to accurately model differences between the different phonetic classes - irrespective of the speaker. Consequently, the idea in VTLN is to extract acoustic observations in such a way that the feature

vectors for a particular phonetic class look similar for different speakers. The method 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 it would eliminate the inter-speaker source of variability in the acoustic feature vectors. The scaling of the frequency axis is assumed to be piecewise linear [18], and is parametrized 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. The procedure for normalizing the feature vectors for a given speaker are as follows: - 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 now 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 a supervised manner using the transcriptions produced from a speaker-independent system to label the phonetic class for each feature vector, and using the corresponding pdf of the canonical model to compute the probability density of the feature vector. Subsequently, the warp scale that gives 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 speaker.

A.1 Results

The experimental results obtained with VTLN adapatation are shown in Table VI – it may be seen that VTLN provides a relative improvement of 7.6%-7.9% in the original cepstral space, and 13.6%-8.2% when the VTLN feature vectors are projected with an MLD projection.

System	Feature space	E-VM1
VM1a	Mel Cepstra	40.5%
VM1c	Mel Cepstra + VTLN	37.4%
VM1d	Mel Cepstra + VTLN + MLD Proj	35.0%
VM2a	PLP Cepstra	40.3%
VM2c	PLP Cepstra + VTLN	37.1%
VM2d	PLP Cepstra + VTLN + MLD Proj	37.0%

TABLE VI
CONSENSUS WORD ERROR RATE FOR VTL NORMALIZED FEATURES

B. Linear Transform Adaptation

The VTLN procedure described earlier tries to map the feature vectors of the training and test speakers into a canonical space by warping the frequency axis in a speaker-dependent manner. However, this simple warping method is not sufficient to eliminate all speaker-dependent variation and produce a truly canonical feature space. Therefore, when the test speaker's data is warped using the same method, the procedure is only a crude approximation to reducing the discrepancy between the test speaker's feature vectors and the acoustic model. As a result, there is sufficient justification to investigate other methods that try to reduce the mismatch between the test speaker's feature vectors and the acoustic model even further.

The next step in reducing this mismatch is to hypothesize a feature transformation of the training speakers data that makes the training data (and consequently, the acoustic model) look more similar to the test speaker's data. We investigate an adaptation method that constrains the transformation to be affine [19] and compute the transformation such that the likelihood of the test speaker's adaptation data is maximized by the transformed acoustic model – we refer to this method as feature-space maximum-likelihood linear regression (FMLLR). FMLLR computes a single affine transformation that is applied to the means and covariances of all the gaussians in the acoustic model. The mismatch between the transformed acoustic model and the test speaker's data can be reduced even further by further transforming the means of the acoustic model. This is done using the well-known MLLR [20] method. For the sake of completeness, we will briefly describe the formulation for these two steps next.

Let $p_{l,j}(x_t) = \mathcal{N}(\mu_{l,j}, \Sigma_{l,j})$ denote the pdf of the j^{th} gaussian of the l^{th} class in the VTLN warped feature space, x_t. The FMLLR method hypothesizes a transformation of the VTLN warped training data of the form $y_t = A_{fmllr}x_t$. Consequently, the pdf of the transformed data is given by scaling the original pdf by the Jacobian of the transformation, i.e. $p'_{l,j}(y_t) = p_{l,j}(x_t = A^{-1}_{fmllr}y_t)/|A_{fmllr}|$. The log-likelihood of the adaptation data is now given by

$$L_{A_{fmllr}}(y_1^T) = -\sum_{l,j,t} c_{l,j}(t) * \left[(A^{-1}_{fmllr}y_t - \mu_{l,j})^T\Sigma^{-1}_{l,j}(A^{-1}_{fmllr}y_t - \mu_{l,j}) + |A_{fmllr}|\right] \quad (13)$$

where $c_{l,j}(t)$ denotes the occupancy count for the j^{th} gaussian of the l^{th} class at time t, and y_t denotes the adaptation data from the test speaker. The objective function (13) has no closed form solution, consequently, gradient descent based optimization methods have to be used to solve for A_{fmllr}. Note that after this transformation has

[3] One of the subtleties associated with the selection of the warp scale is related to the normalization of the likelihoods obtained for different warp scales. The VTLN warping can be thought of as a mapping from the original feature vector x_t to 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 automatically 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 the speakers data, and normalizing by the determinant of this covariance matrix.

been computed, the adaptation procedure can be implemented by transforming the feature vectors of the test speaker $z_t = A_{fmllr}^{-1} y_t$, rather than as a transformation of the acoustic model [4].

Once the adaptation data has been transformed by A_{fmllr} to yield the feature vectors $z_t = A_{fmllr}^{-1} y_t$, the MLLR method now hypothesizes a further transformation of the means of the acoustic model to bring the model closer to the adaptation data. The final transformed models are given by $p''_{l,j}(z_t) = \mathcal{N}(\mu''_{l,j}, \Sigma_{l,j})$ where $\mu''_{l,j} = A_k \mu_{l,j} + b_k$. Several sets of transformations A_k, b_k could be computed depending on how much adaptation data is available. These affine transformations are shared (tied) by a subset of the gaussians in the acoustic model – the tying structure is determined by the distance between the gaussians in the acoustic model. In other words, the gaussians that are close together are likely to share an affine transformation. Further details are given in [20]. The log-likelihood of the adaptation data can now be expressed as

$$L_{A_k,b_k}(z_1^T) = \sum_{l,j,t} c_{l,j}(t) * \log\left[(z_t - A_k\mu_{l,j} - b_k)^T \Sigma_{l,j}^{-1}(z_t - A_k\mu_{l,j} - b_k)\right] \quad (14)$$

By setting $\frac{L_{A_k,b_k}(z_1^T)}{d\underline{a}_{k,d}} = 0$ where $\underline{a}_{k,d}$ denotes the d^{th} row of the matrix $\underline{A_k} = [A_k\ b_k]$, we obtain

$$\sum_{l,j,t}\left[\frac{c_{l,j}(t)}{\Sigma^2_{l,j,d}}\mu_{l,j}\mu_{l,j}^T\right] a_d = \sum_{l,j,t}\left[\frac{c_{l,j}(t) z_{t,d}}{\sigma^2_{l,j,d}}\right]\mu_{l,j} \quad (15)$$

B.1 Results

The results of applying these adapation methods is shown in Table VII. It may be seen that the FMLLR+MLLR adaptation provides a relative improvement of 2%-4% over and above the improvements provided by VTLN warping.

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 [17], a method was proposed whereby the output of multiple ASR systems was treated as independent sources of knowledge, and the outputs were combined to produce a composite output that had a lower WER than any of the individual ASR outputs. The method is based on taking multiple ASR outputs and aligning them to form a word transition network (WTN) for the utterance. This is done by an iterative process whereby the first two ASR system hypotheses are aligned to form a WTN; subsequently the next ASR hypothesis is aligned to the WTN, and so on. Finally, the composite WTN is searched by a voting or scoring module to select the best scoring word sequence.

[4] This is possible because only one transformation A_{fmllr} is shared by all the gaussians in the system. If the method is extended to computing multiple transformations, then there is an additional normalization step that is required before the log-likelihoods of the transformed features can be computed.

System	Feature space	E-VM1
VM1d	Mel Cepstra + VTLN + MLD Proj	35.0%
VM1e	Mel Cepstra + VTLN + MLD Proj + FMLLR+MLLR	34.3%
VM2d	PLP Cepstra + VTLN + MLD Proj	37.0%
VM2e	PLP Cepstra + VTLN + MLD Proj + FMLLR+MLLR	35.5%

TABLE VII
Consensus Word error rate for FMLLR+MLLR Adaptation

	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 VIII
Word error rates with system combination

A. Results

We implemented the ROVER system combination technique to combine the outputs of the various systems that we experimented with. We applied the combination technique to both the MAP hypothesis produced by the ASR system, as well as the consensus hypothesis of Section V. The results are given in Table VIII - and show a consistent relative improvement of 4.3%-4.6% for the MAP and consensus hypothesis case by combining the outputs of the two systems.

VIII. Effect of Training Data

All the experimental results we have reported so far were on systems that were trained on the T-VM2 database. In order to understand the effect of the amount of training data on the system performance, we also experimented with systems that were trained on the complete Voicemail corpus T-VM1 comprising 114 hours of speech, and the subsets T-VM3 comprising 15 hours of speech, and T-VM4 comprising 30 hours of speech. The results are summarized in Table IX, and shown plotted in Fig. 4. Both the speaker-independent WER appear to be almost a linear function of the amount of training data - with additional data likely to help more in the case where the baseline is trained on a small amount of data.

System	Feature Space	E-VM1	E-VM2
VM1b	Mel Cepstra + MLD Proj	39.1%	38.0%
VM1b_s	Mel Cepstra + MLD Proj	47.8%	47.0%
VM1b_m	Mel Cepstra + MLD Proj	41.9%	40.9%
VM1b_l	Mel Cepstra + MLD Proj	30.2%	30.1%
VM1e	Mel Cepstra + VTLN + MLD Proj + FMLLR + MLLR	34.9%	33.7%
VM1e_s	Mel Cepstra + VTLN + MLD Proj + FMLLR + MLLR	43.3%	42.7%
VM1e_m	Mel Cepstra + VTLN + MLD Proj + FMLLR + MLLR	38.1%	37.0%
VM1e_l	Mel Cepstra + VTLN + MLD Proj + FMLLR + MLLR	27.9%	27.2%

TABLE IX
MAP WORD ERROR RATES AS A FUNCTION OF TRAINING DATA

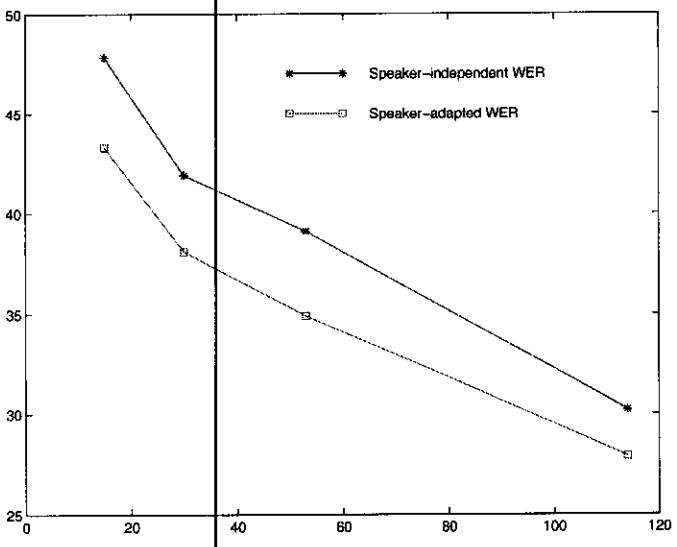

Fig. 4. MAP WER as function of Training Data

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 in 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, combination of hypotheses, and speaker adaptation of acoustic models and were instrumental in reducing the word error rate on Voicemail data to around 27.9%.

More specifically, the algorithms and their relative contributions were :

- a lexicon design technique that yields up to a 3.6% relative improvement in performance
- a novel linear projection (MLD) that can be applied on the extracted features, and yields up to a 5.4% relative improvement in performance. We also show that these methods are better applied on features that have already been canonicalized to eliminate inter-speaker variation
- use of a consensus hypothesis algorithm that yields up to a 3.5% relative improvement in performance
- VTLN adaptation provides up to 4.1% relative improvement in performance
- adaptation by linear transforms FMLLR+MLLR provides up to a 4% relative improvement in performance
- system combination methods that yields up to a 4.7% relative improvement in performance
- MFCC speaker independent WER of 30.3% and speaker adapted WER of 27.7%
- speaker adaptation provides a relative improvement of 9.4% for systems trained on 15 hours, and 7.6% for systems trained on 114 hours - this is consistent with the common wisdom that performance improvements due to adaptation reduce as the system is trained on more data
- WER performance as a function of training data shows that the WER is almost linearly related to the amount of training data [6]

X. ACKNOWLEDGEMENT

We would like to acknowledge the help of Bhuvana Ramabhadran, Stan Chen, Geoffrey Zweig and other members of the Speech Group for several useful discussions.

XI. APPENDIX

The Voicemail transcription task started at IBM in 1997, motivated by the fact that voicemail represents a large amount of real world spontaneous speech with potential applications such as message summarization and information extraction; however, the performance of automatic speech recognition systems on this type of speech does not appear to have been the focus of any previous research efforts. Further, though several speech databases exist for the purpose of carrying out research in speech recognition (Wall Steet Journal, Switchboard/Callhome [1], Hub 4 [2], ATIS) voicemail speech is not well represented in any existing database.

A. Creation of a voicemail database

Voicemail data is unfortunately fairly difficult to collect because of privacy and legal issues. We were able to surmount some of these difficulties by collecting the data

[6] This is contrary to the common wisdom that the WER is dependent on the log of the amount of training data. We do see the slope of the linear function is steeper for lower amounts of training, consequently, our conjecture is that the exponential relationship holds only for much larger amounts of training data.

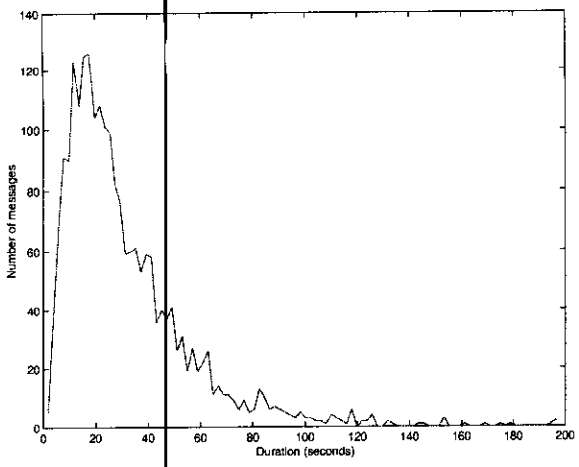

Fig. 5. Histogram of length of voicemail messages

locally at various IBM sites within the US. We adopted the following strategy to build up a database. Volunteers at these sites were asked to forward some of their voicemail messages to a local extension number set up for the purpose of collecting this data. The messages would then be collected periodically from the voicemailbox of this local extension and added to the database. The volunteers were asked to forward only those messages that they felt were non-confidential and if neither they nor the person leaving the message had any objection to its being included in a database. Further, the volunteers were asked to add a sentence to their outgoing message of the form - 'Your voicemail may also be used for commercial research in developing algorithms for speech recognition. If you do not want your data to be used, please say so in your message.' - in order to let the caller know that their message may be included in the database, and to give them an opportunity to decline having their voicemail added to the database. Finally, the volunteers were provided with some incentives for every few messages that they forwarded. Details of the data collection strategy are given in [21]. A subset of this data, comprising of 15 hours of speech, has been made available to the research community through the Linguistic Data Consortium (LDC98S77 - Voicemail Corpus Part I). We intend to make an additional 15 hours of speech in the near future.

A.1 Analysis of voicemail data

Here we give an analysis of a subset of the Voicemail data comprising around 2200 voicemail messages, totaling 19.4 hours of speech. The corresponding transcriptions of these messages had 220K words, and the size of the vocabulary corresponding to these transcriptions was 9.7K words. Examining the histogram of message durations (Fig 5) indicates a peak of 18s and an average of 31s. The database was not entirely gender balanced because we did not do anything explicitly to ensure that this was the case. Approximately 38 % of the messages corresponded to male speakers.

We also did a subjective analysis of the topics covered by the messages, and found that the topics ranged from personal messages to extremely technically oriented messages. In this sense, the Voicemail database is again different from the Switchboard database where the speakers were asked to talk about a specified topic (one of 35 topics), and gives a distribution of topics in real-world voicemail. We attempted to subjectively characterize the topics into (a) business-related (eg. schedule for a meeting), (b)personal (eg. 'get back home before 9 pm ... or else' variety), (c) work-related (eg. maintenance schedule for a lab), (d) technical (eg. 'theres yet another bug in this code ...' variety), and (e) miscellaneous (messages not falling in any of the above categories). Based on a subjective categorization, we found that the percentage of these categories respectively was 27, 25, 17, 13 and 18 % respectively.

REFERENCES

[1] Proceedings of the NIST Speech Transcription Workshop, May 2001.

[2] Proceedings of the DARPA Broadcast News Transcription Workshop, Feb 2000, Morgam Kaufman Publishers.

[3] J. Huang et al., "Performance improvement in Voicemail Transcription", Proceedings of DARPA Speech Transcription Workshop, May 2000.

[4] L. R. Bahl et al., "Performance of the IBM Large Vocabulary Speech Recognition System on the ARPA Wall Street Journal task", Proceedings of ICASSP, 1995.

[5] P. S. Gopalakrishnan, L. R. Bahl and R. L. Mercer, "A Tree Search Strategy for Large Vocabulary Continuous Speech Recognition", Proceedings of ICASSP 1995.

[6] G. Saon and M. Padmanabhan, "Data-driven Approach to Designing Compound Words for Continuous Speech Recognition", IEEE Trans. Speech and Audio Processing, May 2001.

[7] M. Finke and A. Waibel, "Speaking mode dependent pronunciation modeling in large vocabulary conversational speech recognition", Proceedings of Eurospeech 1997, vol. 5, pp 2379-2382.

[8] S. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", IEEE Trans. Acoustics, Speech and Signal Processing, Aug 1980.

[9] H. Hermansky, "Perceptual Linear Prediction of Speech", Journal of the Acoustic Society of America, Apr 1990.

[10] K. Fukunaga. Introduction to statistical pattern recognition. *Academic Press*, New York, 1990.

[11] N. Kumar and A. G. Andreou. Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition. *Speech Communcation*, 26:283–297, 1998.

[12] G. Saon, M. Padmanabhan, R. Gopinath, S. Chen, "Maximum likelihood discriminant feature spaces", Proceedings of ICASSP, 2000.

[13] R. A. Gopinath. Maximum likelihood modeling with gaussian distributions for classification. *Proceedings of ICASSP'98*, Denver, 1998.

[14] M. J. F. Gales. Semi-tied covariance matrices for hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 7:272–281, 1999.

[15] A. Stolcke, Y. Konig, and M. Weintraub, "Explicit word error minimization in N-best list rescoring", Proceedings of Eurospeech, 1997.

[16] L. Mangu, E. Brill and A. Stolcke, "Finding consensus among words: lattice-based word error minimization", Proceedings of Eurospeech, 1999.

[17] J. G. Fiscus, "A Post-Processing System to Yield Reduced Word Error Rates: Recogniser Output Voting Error Reduction (ROVER)", Proceedings of IEEE ASRU Workshop, pp. 347-352, Santa Barbara, 1997.

[18] S. Wegman, D. McAllaster, J. Orloff, and B. Peskin, "Speaker normalization on conversational telephone speech", Proceedings of ICASSP, 1997.

[19] M. J. F. Gales. Maximum Likelihood Linear Transforms for HMM-based Speech Recognition *Computer, Speech and Language*, vol. 12, 1998.

[20] C. J. Legetter and P. C. Woodland, "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models", Computer, Speech and Language, vol. 9, pp 171, 1995.

[21] M. Padmanabhan et al., "Issues involved in Voicemail data collection" Proceedings of the DARPA Hub4 Broadcast News Workshop, 1997.