A Robust High Accuracy Speech Recognition System For Mobile Applications

Sabine Deligne, Satya Dharanipragada, Ramesh Gopinath, Benoit Maison, Peder Olsen, Harry Printz

Abstract

This paper describes a robust, accurate, efficient, low-resource, medium-vocabulary, grammar-based speech recognition system using Hidden Markov Models for mobile applications. Among the issues and techniques we explore are improving robustness and efficiency of the front-end, using multiple microphones for removing extraneous signals from speech via a new multi-channel CDCN technique, reducing computation via silence detection, applying the Bayesian information criterion (BIC) to build smaller and better acoustic models, minimizing finite state grammars, using hybrid maximum likelihood and discriminative models, and automatically generating baseforms from single new-word utterances.

I. Introduction

The proliferation and widespread use of mobile devices in everyday life has brought about a great need for efficient and easy-to-use interfaces to these devices. Technological advances are allowing these mobile devices to be smaller, cheaper, more powerful and more energy-efficient. Diminishing dimensions have made the traditional keyboard/stylus interface of limited use and applicability in mobile applications. A conversational speech-based interface, being relatively device-size independent, is hence emerging as a powerful and viable alternative (and in some cases the only choice) in such applications. The growing necessity of the conversational interface demands significant advances in processing power on the one hand, and speech and natural language technologies on the other. In particular, speech recognition being a key component of the conversational interface, there is significant need for a low-resource speech recognition system that is robust, accurate, and efficient.

This paper describes techniques for reducing the error rate, memory footprint and computational bandwidth requirements of a grammar-based, medium-vocabulary speech recognition system, intended for deployment on a portable or otherwise low-resource device. By medium-vocabulary we mean about 500 distinct words or phrases, possibly constrained within a finite-state grammar. By low-resource we mean a system that can be executed by a 50 DMIPS processor, augmented by 1 MB or less of DRAM. Such a combination of resources is now attainable at a modest price, and can be powered by a battery with an acceptable lifetime [9].

Dr. Harry Printz is currently with AgileTV Corporation. The rest are with the IBM Watson Research Center Yorktown Heights, NY 10598 USA, Contact: dsatya@watson.ibm.com

This objective is both appealing in its promise and daunting in its technical challenges. Such systems are by design highly portable; for this reason they are taken everywhere, and so high accuracy in adverse acoustic environments is an important issue. Moreover because they are intended for the mass market, the hardware must be low cost. Likewise due to the desire to appeal to the consumer, the system must not require a complicated enrollment procedure, yet it must offer the user the ability to easily personalize it to his/her needs.

The paper is organized as follows. We begin with a general overview of the system organization in Section II. Section III describes the front-end of the system. We then describe the acoustic models in Section IV. In particular we address training and model size in this section. Section V describes techniques for grammar minimization. In sections III-B and VI-A we describe two features of the system that address the portability and robustness of the system. Section III-B describes a new multi-channel CDCN technique for removing extraneous signals from speech using multiple microphones. Section VI-A describes a feature that permits new words to be added to the vocabulary on the fly. We conclude with a summary and discussion of the potential for low-resource speech recognition.

II. System Organization

The system uses the familiar phonetically-based, hidden Markov model (HMM) approach. Logically it is divided into three primary modules: the front-end, the labeler and the decoder as shown in Fig. 1. When processing speech, the computational workload is divided approximately equally among these modules. However the front-end may be active more than the other modules, since as we describe below it is used as well to separate speech from non-speech audio.

The front-end computes standard 13-dimensional mel-frequency cepstral coefficients (MFCC) from 16-bit PCM sampled at 11.025 KHz. The front-end also performs adaptive mean and energy normalization. Section III describes the front-end processing in more detail.

The labeler computes first and second differences of the 13-dimensional cepstral vectors, and concatenates these with the original elements to yield a 39-dimensional feature vector. The labeler then computes the log likelihood of each feature vector according to observation densities associated with the states of the system's HMMs. This computation yields a ranked list of the top 100 HMM states. Likelihoods are inferred based upon the rank of each HMM state by a table lookup [5]. The sequence of rank likelihoods is then forwarded to the decoder.

The decoder implements a synchronous Viterbi search over its active vocabulary, which may be changed dynamically. Words are represented as sequences of context-dependent phonemes, with each phoneme modeled as a three-state HMM. The observation densities associated with each HMM state are conditioned upon one phone of left context and one phone of right context only. Each observation density is modeled as a mixture

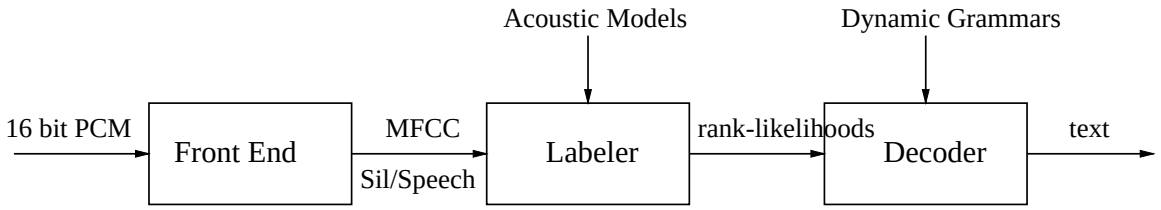


Fig. 1. Flow Chart of the System Architecture.

of 39-dimensional diagonal Gaussians.

III. Front-End Processing

Speech samples are partitioned into overlapping frames of 25 ms duration with a frame-shift of 15 ms. A 15 ms frame-shift instead of the standard 10 ms frame-shift was chosen since it reduces the overall computational load significantly without affecting the recognition accuracy. Each frame of speech is windowed with a Hamming window and represented by a 13 dimensional MFCC vector. We empirically observed that noise sources, such as car noise, have significant energy in the low frequencies and speech energy is mainly concentrated in frequencies above 200 Hz. The 24 triangular mel-filters are therefore placed in in the frequency range [200Hz – 5500 Hz], with center frequencies equally spaced in the corresponding mel-frequency scale. Discarding the low frequencies thus, improves the robustness of the system to noise.

The front-end also performs adaptive mean removal and adaptive energy normalization to reduce the effects of channel and high variability in the signal levels respectively. The cepstral mean removal ameliorates channel variability and consists in subtracting from each incoming frame, $\mathbf{c}_t$, the current estimate of the mean value of the cepstra $\overline{\mathbf{c}}_t$, for the current utterance;

$$\tilde{\mathbf{c}}_t = \mathbf{c}_t - \overline{\mathbf{c}}_t \tag{1}$$

The cepstral mean is initialized with a value estimated off line on a representative collection of training data. It is then updated on a per-frame basis by interpolating the current mean estimate with each incoming frame;

$$\overline{\mathbf{c}}_t = \lambda \overline{\mathbf{c}}_{t-1} + (1-\lambda)\mathbf{c}_t \tag{2}$$

The interpolation weights are determined based upon the energy level of the incoming frame (low energy frames, that are closer to silence, do not contribute to the update). The energy normalization consists of subtracting from the zeroth dimension of each incoming frame, $\mathbf{c_t}(0)$, an estimate of the maximum of the zeroth cepstral coefficient, $\hat{\mathbf{c}}_\mathbf{t}(0)$;

$$\tilde{\mathbf{c}}_t(0) = \mathbf{c_t}(0) - \hat{\mathbf{c}}_\mathbf{t}(0). \tag{3}$$

$\hat{\mathbf{c}}_\mathbf{t}(0)$ is larger of the maximum c_0 value observed up to the t^{th} frame of the n^{th} utterance and the maximum c_0 value observed during the $(n-1)^{\text{th}}$ utterance.

A. Speech / Silence Detection

In addition to computing MFCC vectors, the front-end separates speech from silence. Using simple Gaussian mixture models, the front-end labels each cepstral vector as speech or silence, and buffers these vectors for later processing. We use a mixture of 4 Gaussians each to model the distribution of silence frames and the distribution of speech frames. All Gaussians have diagonal covariances. They are estimated on a balanced collection of quiet and noisy data previously labeled with speech and silence labels. When a sufficiently long sequence of vectors labeled as speech has accumulated in the buffer, the front-end decides that it is receiving spoken language for decoding, and forwards the accumulated sequence of vectors (and those that it continues to generate) to the downstream modules for decoding. Sequences of vectors that are classified as silence are discarded without processing by the labeler and the decoder, providing a substantial computational savings.

B. Front-End Robustness Via Multi-Channel CDCN

B.1 Unwanted signal removal

Robustness in the presence of noise, and more generally in the presence of interfering signals, is a crucial issue for speech recognition to work in a real-world environment. In cases where the signal interfering with the speech is stationary, depending on whether or not the characteristics are known in advance, robustness issues can, to a certain extent, be addressed with multi-style training or spectral-subtraction/CDCN [17]. However, in most applications, the signal corrupting the speech is neither known in advance nor stationary (for example, music or speech from competing speakers). Such cases cannot be handled by devising special training schemes and they require the use of on-line adaptation algorithms. In this section, we address the case where recordings of the interfering signals are available in separate channels. These signals are called the reference signals. This occurs for example when the speech signal is corrupted by the sound emitted by a radio or a CD player (the reference signals are recorded at the outputs of the radio or CD player), or, when the speech signal is mixed with the speech of competing speakers (the reference signals are recorded from the microphones of the competing speakers). The general problem of removing unwanted signals from a desired signal by using reference signals is typically addressed with adaptive decorrelation filtering techniques [15]. In decorrelation filtering, the corrupted signal and the reference signal are assumed to be observed at the output of a linear system modeling the cross-coupling between the desired signal and the interfering signal. This linear system is assumed to be such that there is no leakage of the desired signal into the reference signal. Further assuming that the desired and interfering signal are uncorrelated, the linear system can be estimated unambiguously so that the desired signal can be recovered via inverse filtering. However adaptive decorrelation filtering suffers from some limitations in the context of speech recognition, especially in the context of embedded applications running with limited computational resources: (i) it performs in the waveform domain, on a sample basis, thus leading to a high computation rate, (ii) it involves an iterative scheme, hence some delay may occur before it

converges towards an accurate estimate of the coupling system, especially in a non-stationary environment, (iii) its performance depends on the modeling accuracy of the coupling system (the length of the decorrelating filters needs to be hypothesized). We present an approach especially designed to deal with a real time application constrained to run with low computational resources. An inexpensive - and inaccurate - form of adaptive filtering, assuming a single-tap delay filter, is used to roughly align and scale the reference signal with the noisy speech. The aligned and scaled reference signal is then removed from the noisy speech in the cepstral domain by using our new algorithm derived from CDCN [17] and called MCDCN: Multi-channel Codebook Dependent Cepstral Normalization. As will be shown in this paper, MCDCN is advantageous as: (i) it allows to compensate for the loose modeling of the coupling system between the speech and the interfering signal by taking advantage of our knowledge of the clean speech distribution in the cepstral domain, (ii) it does so through the use of a codebook, the size of which can be adjusted to meet the desired balance between performance and computational complexity, (iii) it performs on a per-frame basis, i.e. at a low computation rate compared to waveform techniques (every 165 samples with our 15ms system on 11kHz data, instead of every sample), (iv) it does not involve any iterative estimation scheme, thus further enabling a real time use.

B.2 Principle of MCDCN

MCDCN refers to a multi-channel version of CDCN that allows to compensate for non-stationary noise in cases where the source(s) of noise are recorded separately. In the standard CDCN framework, the desired speech signal is assumed to be first passed through a linear filter, which models the effect of the channel, and then corrupted with noise. In this paper, only the cepstral distortion caused by the noise is considered: the effect of the channel is assumed to be compensated for by the preliminary alignment and scaling explained in section III-B.3. Thus, assuming additive uncorrelated noise, the relation between the power spectral densities of the clean speech, $P_y(f)$, of the noisy speech, $P_x(f)$, and of the noise corrupting the speech, $P_n(f)$, is:

$$P_y(f) \;=\; P_x(f) \;-\; P_n(f) \tag{4}$$

Note that equation 4 suggests that, given the corrupted speech and the noise observed in the reference channel, the spectrum of the clean speech could be recovered with spectral subtraction. However our preliminary experiments tend to indicate that this would require to identify the cross-coupling system between the noise and the clean speech, so that the noise actually corrupting the speech can be estimated by filtering the noise in the reference channel. In [16], adaptive lattice-ladder filters are used to very accurately align the reference signal with the noise present in the corrupted speech. The aligned reference signal is then removed from the noisy speech by using a spectral subtraction technique. The MCDCN technique presented in this paper allows to avoid the cost of an accurate alignment: the imprecision of our estimate of the corrupting noise, as well as the imprecision of the additive noise model, is compensated for by taking advantage of our *a priori* knowledge of the clean speech distribution in the cepstral domain. Besides, MCDCN does not require any

empirical tuning whereas spectral subtraction requires to define an adequate flooring of the cleaned spectrum. The relation between the cepstral vectors of the clean speech $y(t)$, the noisy speech $x(t)$ and the noise $n(t)$ can be expressed as [17]:

$$y(t) \;=\; x(t) \;-\; r(y(t), n(t)) \tag{5}$$

with r a non linear function of both the clean speech and the noise. Assuming MFCC vectors computed with a bank of Mel-filters followed by a Discrete Cosine Transform:

$$r(y(t), n(t)) \;=\; DCT\ \log(1 \;+\; e^{DCT^{-1}\,(n(t)\,-\,y(t))}) \tag{6}$$

where DCT and DCT^{-1} refer respectively to the Discrete Cosine Transform and to its inverse. Whereas in standard CDCN, the noise is estimated via an EM algorithm, we propose in MCDCN to compute the cepstra $n(t)$ of the noise from the reference signal which is assumed to be recorded in a separate channel. For lack of knowing the cepstra $y(t)$ of the clean speech, the function r, like in standard CDCN, is approximated with its expected value over y, given $n(t)$ and $x(t)$:

$$\hat{r}(y(t), n(t)) \;=\; E_{y(t)}[r(y(t), n(t)) \mid x(t), n(t)]$$

To simplify the computation, the function $r(y(t), n(t))$ is assumed to be a piece-wise constant function of $y(t)$. Therefore, assuming a codebook $C_{n_C} = \{\ c_i\ \}_{i=1}^{n_C}$ of n_C cepstral vectors describing the acoustic space of the clean speech, the noise correction term is computed as:

$$\hat{r}(y(t), n(t)) \;=\; \sum_{i=1}^{n_C} p(c_i \mid x(t), n(t))\ r(c_i, n(t)) \tag{7}$$

Assuming the Gaussian distribution $\mathcal{N}(y(t)\ ;\ \mu_i\ ,\ \sigma_i^2)$ to model the distribution of the clean speech $y(t)$ given the codeword c_i, we approximate the distribution of the noisy speech $x(t)$, given c_i, with the Gaussian distribution $\mathcal{N}(x(t)\ ;\ \mu_i + r(c_i, n(t))\ ,\ \sigma_i^2)$. The posterior probability of the codeword c_i, given $x(t)$ and $n(t)$, is thus computed as:

$$p(c_i \mid x(t), n(t)) = \frac{\pi_i \mathcal{N}(x(t); \mu_i + r(c_i, n(t)), \sigma_i^2)}{\sum_{j=1}^{n_C} \pi_j \mathcal{N}(x(t); \mu_j + r(c_j, n(t)), \sigma_j^2)}$$

where π_i denotes the *a priori* probability of the codeword c_i. In [18], CDCN is used with a more refined estimation of the distribution of the noisy speech, inspired by the model combination framework. An estimate of the clean speech $\hat{y}(t)$ is computed as:

$$\hat{y}(t) \;=\; x(t) \;-\; \hat{r}(y(t), n(t)) \tag{8}$$

The computational cost of MCDCN is a linear function of the size n_C of the codebook. The codebook can thus be designed so as to find the desired balance between performance and computational complexity

B.3 Evaluation of MCDCN

To collect the evaluation data, 20 subjects (10 males and 10 females) were given 50 sentences consisting of digit strings or command phrases. Each subject was asked to repeat the 50 sentences in a stationary car with the speakers playing either radio news or CD music (opera, DJ or jazz music) at 3 signal power levels: 60 dB, 70 dB and 80 dB in average, as measured by an SPL meter between the front seats at about lap level. The speech corrupted by the sound emitted by the car speakers was recorded with an AKG Q400 microphone located on the visor. Simultaneously, the signals at either the radio output or at the left and right outputs of the CD player were captured in separate channels. All the data were recorded at 22kHz and downsampled to 11kHz.

In the experiments presented here, the interfering signal comes from either a mono source (the radio of the car) or a stereo source (the CD player of the car). In the stereo case, the signals from the left and right outputs of the CD player are aligned in turn against the waveform of the noisy speech. The aligned left and right waveforms are then summed up so that only one reference signal is actually used when applying MCDCN. The nosiy speech and reference waveforms are aligned by detecting the maximum of their cross-correlation function for shifts of up to 90ms. They are scaled by estimating the mean ratio of the signal in each channel during the first 450ms of the recording (we assumed that there is no speech during the first 450ms of each sentence). This preliminary step represents a simple form of filtering ; a more refined scheme like adaptive decorrelation filtering for example could be used instead, but at a much higher cost.

After alignment and scaling, the cepstra in the reference channel and in the noisy speech channel are used as estimates of respectively $n(t)$ and $x(t)$ in equations 7 and 8. MCDCN is applied by using codebooks of either 2, 4, 8, 16, 32, 64, 128 or 256 codewords. Each codebook was estimated by quantizing about 3,000 sentences of clean speech (recorded with the same microphone as the evaluation data) by assuming diagonal covariance matrices tied across all codewords. All codewords were assigned equal priors. Speech recognition is performed on the estimate $\hat{y}(t)$ of the cepstra of the clean speech obtained from equation 8.

Table I shows the average word error rates (WER) obtained by decoding the noisy speech without using any compensation and by compensating with MCDCN with codebooks of size ranging from 2 to 256 codewords. The average is taken over all the speakers and all the interfering signals (radio and all music styles) at each of the three sound levels. With as few as 2 codewords, MCDCN allows a relative WER reduction of about 75% with the 60 and 70dB interferences, and 65% with the 80dB interferences. The best performance at 60 dB corresponds to an 82% WER reduction and it necessitates codebooks of at least 8 codewords. The best performance at 70 dB and 80dB corresponds to WER reductions of respectively 87% and 76% with codebooks of at least 32 codewords. Our experiments tend to indicate that the minimal number of codewords required to reach an optimal performance is in relation with the power level of the interfering noise: the louder the noise, the bigger the codebook needs to be. Our interpretation is that the approximation used in the CDCN

framework, according to which $r(y(t), n(t))$ is a piece-wise constant function of $y(t)$, holds better at low levels of noise[1]. The histograms on figures 2 show the WER when the interfering signal is a) the opera, b) the DJ music, c) the jazz music and d) the radio news talk. The most confusing interference for the speech recognition system is the competing speech from the radio speaker, and then the DJ kind of music (note that the DJ tracks consisted of mainly rap music, i.e. somehow again a competing speech). The effect of the radio however is better compensated than the effect of the DJ music, possibly because the radio is a mono source and our simple alignment+scaling scheme can better approximate the channel effect than with a stereo source. In our experiments, the delays between the noisy speech waveform and the reference waveforms were

	60dB	70dB	80dB
no compensation	6.1	23.8	44.5
$n_C = 2$	1.5	5.5	15.6
$n_C = 4$	1.2	4.5	13.9
$n_C = 8$	**1.1**	4.1	12.8
$n_C = 16$	**1.2**	3.3	12.0
$n_C = 32$	**1.2**	**3.0**	**10.8**
$n_C = 64$	**1.1**	**3.0**	**10.8**
$n_C = 128$	**1.0**	**3.0**	**10.7**
$n_C = 256$	**1.2**	**3.0**	11.5

TABLE I

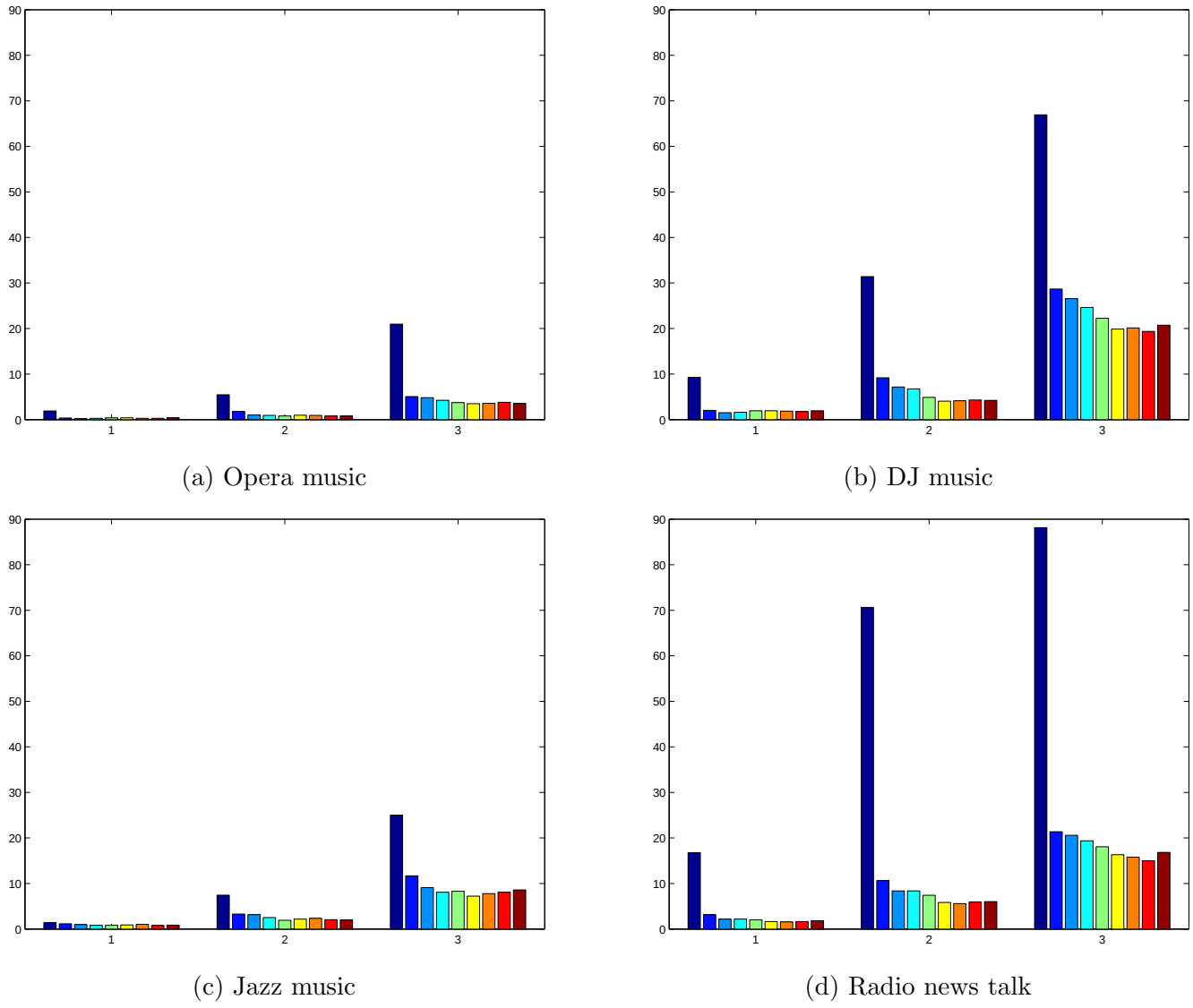
AVERAGE WER OVER ALL SPEAKERS AND INTERFERING SIGNALS, FOR INTERFERING SIGNALS AT POWER LEVEL 60, 70 AND 80DB, AND FOR CODEBOOKS OF VARIOUS SIZES

found to be in the range 5 to 15 ms. We tried to alleviate the possible impact of mis-alignments by applying, within each channel, the cepstral averaging technique presented in [19]: each 15ms cepstrum is obtained by averaging 3 cepstra computed with 5ms shifts. It resulted in about 10% relative improvement at the lowest power level (0.9% versus 1.0%), but it hurt the accuracy at the loudest levels. This is consistent with the fact that mis-alignments are more likely to occur when the amount of interfering signal in the corrupted speech is small.

IV. Acoustic Models

The acoustic model comprises context-dependent sub-phone classes (allophones). The context for a given phone is composed of only one phone to its left and one phone to its right. A key issue is whether or not the phone context is permitted to extend across word boundaries. We have investigated both approaches and

[1]Actually, it can be verified that, for a given $y(t)$, if $\mid n_1(t) \mid \leq \mid n_2(t) \mid$ then $\mid \frac{\partial r(y(t),n_1(t))}{\partial y(t)} \mid \leq \mid \frac{\partial r(y(t),n_2(t))}{\partial y(t)} \mid$.



(a) Opera music

(b) DJ music

(c) Jazz music

(d) Radio news talk

Fig. 2. WER averaged over all speakers for each interfering source (opera,DJ,jazz,radio). The 3 groups of bars on each figure correspond to interfering signals at the power levels 1. (60dB), 2. (70dB) and 3. (80dB). The first bar at each sound level shows the WER when decoding without compensation, and the other bars show the WER when decoding with codebooks of increasing size: 2, 4, 8, 16, 32, 64, 128 and 256 from left to right.

settled on a system that uses within-word context only (that is, context does not extend over word boundaries) since this made the search simpler and faster and had little or no effect in recognition accuracy for the tasks of interest. Except as noted in Section V all results in this paper are for such systems. The allophones are identified by growing a decision tree using the context-tagged training feature vectors and specifying the terminal nodes of the tree as the relevant instances of these classes [5].

Each allophone is modeled by a single-state Hidden Markov Model with a self loop and a forward transition. The training feature vectors are poured down the decision tree and the vectors that collect at each leaf are modeled by a Gaussian Mixture Model (GMM), with diagonal covariance matrices to give an initial acoustic

model. Starting with these initial set of GMMs several iterations of the standard Baum-Welch EM training procedure is run to obtain the final baseline model.

In our system, the output distributions on the state transitions are expressed in terms of the rank of the HMM state instead of in terms of the feature vector and the GMM modeling the leaf. The rank of an HMM state is obtained by computing the likelihood of the acoustic vector using the GMM at each state, and then ranking the states on the basis of their likelihoods.

A. Multistyle Training

We adopt a multistyle training approach for the training the GMMs. We targeted a automotive speech recognition application for testing our system. The training data consists of speech collected in a stationary and moving car at two different speeds – 30 mph and 60 mph. Data was recorded in several different cars with a microphone placed at a few different locations – rear-view mirror, visor and seat-belt. The training data was also appended by synthetically adding noise, collected in a car, to the stationary car data. Overall we have 250 hours of training data. Only the clean (stationary car) data was used to grow the decision tree. The GMMs were trained on the entire data. Overall, we had 680 HMM states in our acoustic model. A total of just over 10,000 Gaussians model all the states.

The test data comprises of 27 speakers recorded in a car moving at speeds 0 mph, 30 mph and 60 mph respectively. Four tasks were considered: addresses (A), commands (C), digits (D) and radio control (R). Following are typical utterances from each task:
A: NEW YORK CITY NINETY SIXTH STREET WEST
C: SET TRACK NUMBER TO SEVEN
D: NINE THREE TWO THREE THREE ZERO ZERO
R: TUNE TO F.M. NINETY THREE POINT NINE
The test set, in total, has over 73,000 words in it.

The performance of the baseline model in each of these tasks and speeds is shown in Table II. The overall performance is tabulated in Table III. We also characterize the performance as a function of the Signal-to-Noise Ratio (SNR) in the utterance. The SNR in a test utterance is computed as:

$$\text{SNR} = 10 \log_{10} \frac{\text{average speech energy}}{\text{average noise energy}} \tag{9}$$

The average speech and noise energies are computed from frames labeled as speech and silence. Figure 4 shows the performance of the baseline model as a function of the SNR.

B. Hybrid ML/MMI Models

The acoustic models of most medium- and large-vocabulary speech recognition systems are trained by maximum likelihood (ML) methods. It is well-known however that for small vocabularies, a discriminative

or maximum mutual information (MMI) approach can yield superior results. The drawbacks of the latter for large vocabularies are the lack of training data for adequate generalization, and the workload of the training computation.

In an attempt to get the best of both worlds, we investigated a hybrid ML/MMI acoustic model. The principle is to estimate distinct acoustic models for the phones used in a selected subset of the full vocabulary, for example digits or letters. In this scheme, the words in the subset (called the target words) are represented by their own phone models, which are not used in the balance of the vocabulary. A discriminative training procedure is applied to estimate the parameters of these phones, while the ML-trained models of the remaining phones remain unchanged. MMI training attempts to simultaneously (i) maximize the likelihood of the training data given the sequence of models corresponding to the correct transcription, and (ii) minimize the likelihood of the training data given all possible sequences of models allowed by the grammar describing the task (including the correct transcription). We adopted the MMI estimation equations described in [11] and [14], using the thresholding scheme proposed in [12]. The mean μ_i and the variance σ_i^2 of the Gaussian i (assuming diagonal covariances) are re-estimated as:

$$\hat{\mu}_i \;=\; \frac{(\theta_i^{num}(\mathcal{O}) \;-\; \theta_i^{den}(\mathcal{O})) + D_i\mu_i}{(\gamma_i^{num} \;-\; \gamma_i^{den}) + D_i} \tag{10}$$

$$\sigma_i^2 \;=\; \frac{(\theta_i^{num}(\mathcal{O}^2) \;-\; \theta_i^{den}(\mathcal{O}^2)) + D_i(\sigma_i^2 \;+\; \mu_i^2)}{(\gamma_i^{num} \;-\; \gamma_i^{den}) + D_i} \;-\; \hat{\mu}_i^2 \tag{11}$$

where $\theta_i(\mathcal{O})$ and $\theta_i(\mathcal{O}^2)$ are respectively sums of data and squared data, weighted by the occupancy counts for the Gaussian i. The superscript num and den refer to the statistics collected for the correct transcription and to the statistics collected for all the transcriptions allowed by the grammar respectively. The value of D_i is set at the maximum of (i) twice the value necessary to ensure positive variance updates for all dimensions of the Gaussian i, and (ii) a global constant multiplied by the denominator occupancy γ_i^{den}.

The advantages of a hybrid ML/MMI scheme are as follows. First, it improves the discriminability of the target words, which are precisely those known to be prone to confusability. Second, the models are trained only on the speech data corresponding to the target words; the data corresponding to non-target words are not used. This is advantageous because it was shown in [13] that minimum classification error training performs better when supplied with well articulated words. Since typically the target words will be content words, as opposed to short function words like prepositions and articles, we may expect them to be reasonably well articulated. Hence their associated phone models may be reliably re-estimated with any discriminative training technique. Third, it makes discriminative training computationally tractable, since it is performed on a subset of the acoustic models and a subset of the training data.

Table II gives experimental results for three acoustic models, respectively baseline, MMI 1 (after one iteration of MMI training) and MMI 3 (after 3 iterations), for test set described above. The overall WER

TABLE II

PERFORMANCE OF ML/MMI HYBRID ACOUSTIC MODEL. EACH COLUMN CONTAINS WER (%) AND SER (%) RESPECTIVELY. SEE TEXT FOR DISCUSSION.

0 mph	A	C	D	R
baseline	2.7 10.0	1.0 2.8	1.2 7.9	0.9 3.2
MMI 1	3.1 11.1	1.0 2.6	1.0 6.5	0.7 2.8
MMI 3	3.2 11.5	1.2 3.0	1.0 6.9	0.8 2.8
30 mph	A	C	D	R
baseline	4.0 13.7	1.7 4.3	3.2 18.8	1.5 6.1
MMI 1	4.0 14.0	1.5 4.2	2.9 17.2	1.5 6.0
MMI 3	4.3 15.0	1.8 4.6	2.9 17.2	1.4 5.5
60 mph	A	C	D	R
baseline	7.9 25.7	6.0 12.9	15.5 55.3	5.4 18.9
MMI 1	7.2 24.1	5.4 11.7	15.3 54.8	5.4 19.0
MMI 3	7.3 24.6	5.5 11.9	14.0 51.2	5.3 18.7

with the MMI 1 and MMI 3 models are 4.62% and 4.50% respectively. Figure 4 shows the performance of the MMI 1 and the MMI 2 models as a function of the SNR.

C. Model Selection

In a resource-constrained system, model size has significant computational and storage consequences. A large model requires more non-volatile storage than a small one, and its associated computations usually require more processor cycles and runtime memory. It is worth noting however that a really sharp and accurate model may permit the subsequent decoding phase to proceed more efficiently.

For these reasons it is desirable to have the smallest possible acoustic model, consistent with the required level of system accuracy. Conversely the technique explored here may be used to generate a superior model (that is, one yielding higher recognition accuracy) at a given fixed size.

We now describe a method for generating acoustic models that are both compact and accurate. The idea is to efficiently deploy model parameters, allocating more parameters to those elements of training data that require them.

We approach this issue by formulating it as a problem of model selection. The problem of model selection is that of picking one model among a set of parametric models. If the models in the set differ in the number of parameters they contain, then training data log likelihood is not by itself a sufficient criterion for choosing among them. For on the one hand models with too few parameters will not adequately represent the data.

But on the other hand models with too many parameters (which presumably have the highest training data log likelihood) will not generalize well to new data. Finding a balance between these extremes of underfitting and overfitting is what model selection is all about.

In speech recognition the most popular methods for model selection are cross-validation and the Bayesian Information Criterion [1], hereafter BIC. We now proceed to investigate the latter. Let $\mathcal{M}$ be an acoustic model, containing n_g Gaussians, with d the dimensionality of the training data; then $M = (2d+1)n_g$ is the total number of parameters needed to describe the model. Let $\mathcal{X}$ be the collection of training data (comprising N points), and let $P(\mathcal{X} \mid \mathcal{M})$ be the training data likelihood. With this notation the BIC penalized likelihood is defined to be

$$BIC(\mathcal{X}, \mathcal{M}) = \log P(\mathcal{X} \mid \mathcal{M}) - \frac{\lambda M \log(N)}{2} \; . \tag{12}$$

The model $\mathcal{M}$ that maximizes $BIC(\mathcal{X}, \mathcal{M})$ is the acoustic model of choice. Strictly speaking, the value $\lambda = 1$ is prescribed in [1], but varying λ allows us to adjust the model complexity in a principled way. Note that the right hand size of equation (12) can be interpreted as $\log\left(P(\mathcal{X} \mid \mathcal{M}) \cdot P(\mathcal{M})\right)$, where $P(\mathcal{M})$ is a prior on model size that prefers small models, with $P(\mathcal{M}) \propto N^{-\lambda M/2}$.

BIC has previously been successfully used for acoustic model selection, clustering, building decision trees and change point detection. The interested reader should consult [2] for more information.

Of interest to us is the application to acoustic model selection. Choosing n_g to maximize (12) for each of the allophones allows those with complicated structure—such as vowels—to be modeled with many Gaussian components, whereas those with simple structure—such as fricatives—can be modeled with few Gaussians [2]. Thus we may deploy parameters more efficiently, allowing more Gaussians to be used for complex sounds, with fewer used for simple sounds.

For $\lambda = 1$, the maximizing n_g of equation (12) is too large for low-resource speech recognition. We explored other values by use of the following strategy. Using the EM algorithm, we trained Gaussian mixtures for each of the 680 states in our system for $n_g = 1, 2, 3, \ldots$, and stored the resulting models and their corresponding likelihoods. This was done using fixed alignments. We were then able to rapidly compute the maximizing n_g for each state for any given value of λ. Choosing a particular λ then determined the BIC-optimal model for each state; the collection of these state models then constitutes a complete acoustic model of a particular size. The resulting decoding accuracy for various model sizes is plotted in Figure 3.

If desired these models may be further trained using variable alignments. Table III shows a comparison between our baseline system built without using BIC, a BIC system built using fixed alignment training with a comparable number of Gaussian mixture components and the same BIC system retrained with variable alignments. Figure 4 shows the performance of the BIC system retrained with variable alignments as a function of the SNR.

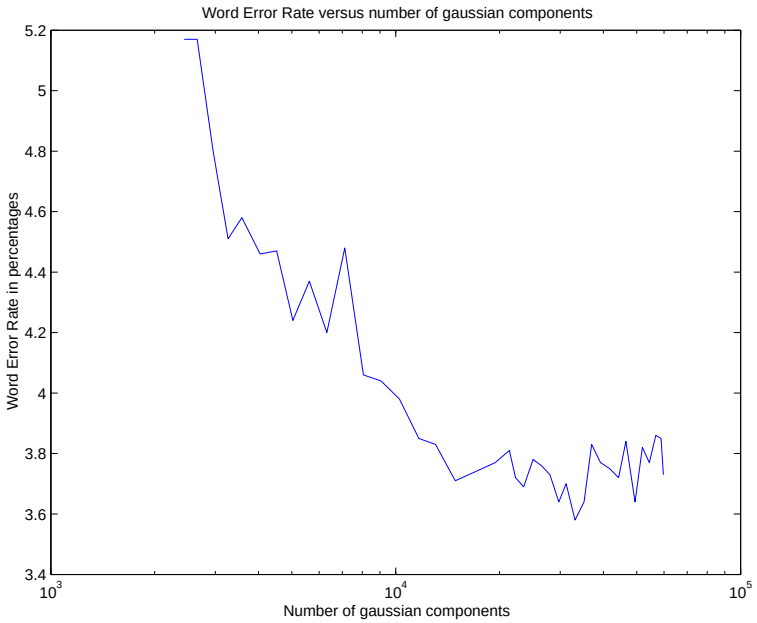


Fig. 3. WER vs n_g for BIC with Fixed Alignments.

TABLE III

WER (%) FOR SYSTEMS BUILT WITH AND WITHOUT BIC.

Acoustic Model	n_g	Overall WER (%)
baseline (no BIC)	10508	4.80
BIC with fixed alignments	10253	3.98
BIC with variable alignments	10253	3.72

V. GRAMMAR MINIMIZATION

A finite-state grammar can be represented as a weighted finite-state automaton on words, where each transition carries a word and a language model probability. A word is mapped to a sequence of phones, each of which in turn is modeled as a three-state HMM. The phone models are context dependent. In this section we only consider phones with cross-word context; that is, the context of the first phone of a word extends backward to the last phone of the word that precedes it. Hence, the first three states of a word model depend on the word that precedes it, as illustrated in Figure 5 (self-loops and transition probabilities are omitted).

A word identifier is present at the end of each sequence of states that models a word. Although they are not strictly necessary, these identifiers allow a fast mapping from the best sequence of states to the recognized words when the search is complete.

Through determinization and minimization, a weighted automaton with a smaller number of states may be created [3]. However, the set of paths through the minimized graph is identical to the set of paths through the original one. Hence, the sequence of states that best explains the acoustic observations is unchanged. In

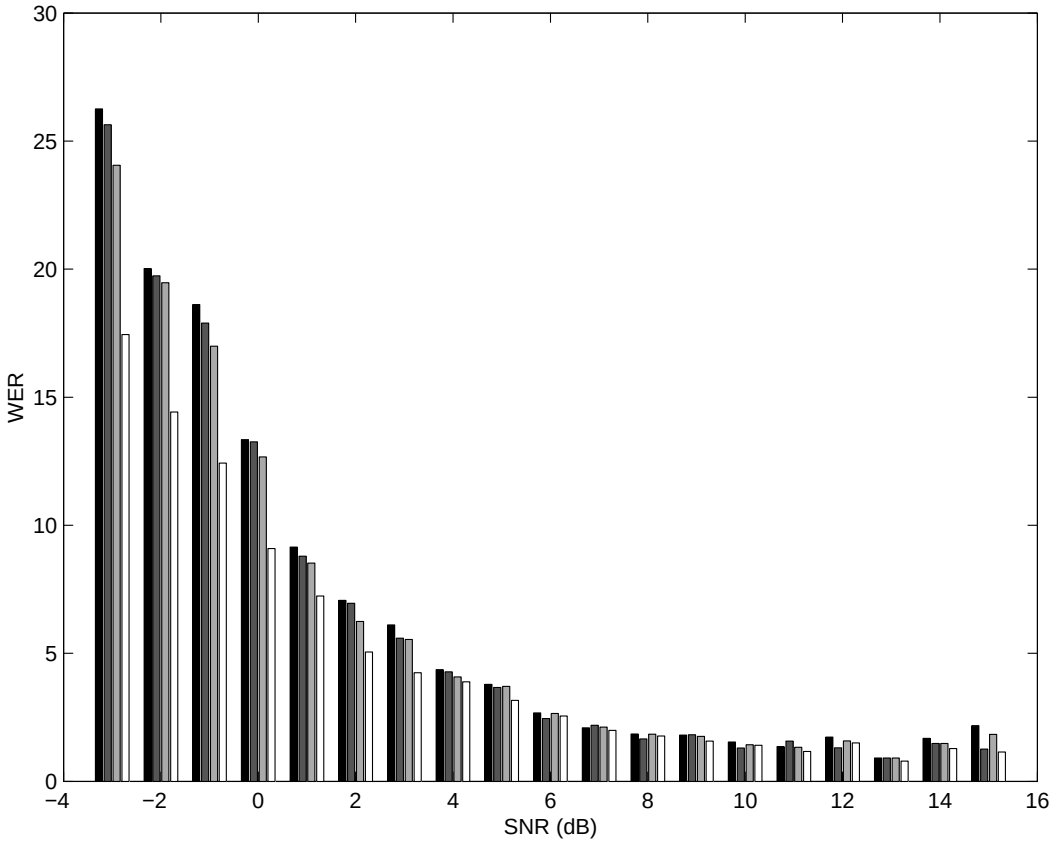


Fig. 4. WER (%) versus SNR for the different acoustic models; bars from left to right baseline, MMI 1, MMI 3, and BIC with variable alignments

other words, if full searches (that is, with no pruning) through both lattices are performed, they will always produce the same results.

Minimization essentially consists of sharing common states between paths that diverge or converge at a graph node (in this case, at word to word transitions). Note that the word identifiers prevent such sharing at the ends of words, and therefore the minimization is not as great as might be achieved, were they absent.

We explored the effects of such minimization on an assortment of grammars. The number of states in the HMM graphs before and after minimization are reported in Table IV. The memory requirements for the storage of the graph and for the search are reduced in the same proportions. As mentioned above, the recognition accuracy is not affected.

Although the number of states to be visited during the search is reduced, the minimized graph is less regular than the original one. This has important and surprising computational consequences. First note that the sharing of states reduces the amount of computation, since identical, parallel paths through the grammar are coalesced into a single path. But the coalesced paths must branch out again, since they ultimately terminate in different words.

This branching-out forces conditional execution to take place in the Viterbi search code, whereas the

Grammar	# states initial	# states minimized
Digits	4611	2651
Commands 1	18685	4657
Addresses	7184	2622
Commands 2	5500	3260
US Phone Numbers	16325	10076
Commands 3	17025	8541

TABLE IV

Sizes of Initial and Minimized HMM Graphs.

unminimized graph consisted of long, unbranching sequences of phones. It is in this sense that the minimized graph is less regular than the original. The unexpected consequence of this reduced regularity is that a typical RISC processor's underlying hardware, which usually includes a deeply pipelined ALU, cannot function at high efficiency. Thus while minimization significantly reduces the total number of arithmetic operations during decoding, it introduces so many pipeline bubbles that only a modest increase in decoding speed is achieved. See [4, Chapter 6] for a discussion of pipelined hardware.

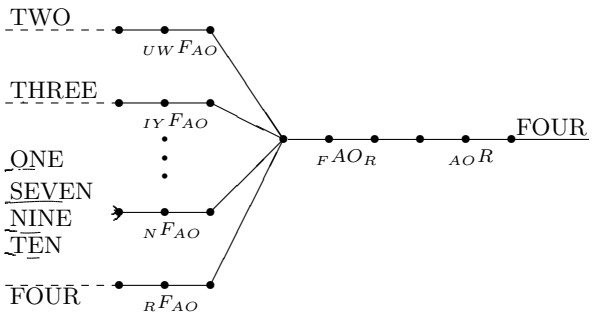


Fig. 5. Subgraph of a Digits Grammar. This figure shows how intra-word context influences graph structure at the start of a word.

VI. Dynamic vocabulary with automatically generated pronunciations

A. Dynamic vocabulary

An additional feature of our system is dynamic vocabulary expansion. This means that the user can add new words to the recognition vocabulary by simply uttering them once or twice. Pronunciations for these words are automatically derived from these utterances, and added to the recognition lexicon. This acoustic-driven approach allows the use of dynamic vocabularies even in small devices that have no keyboard, since it does not require entering the spelling of the new words. Moreover, even if such an interface were to be available, the spellings may not be of very much help as these applications typically involve words the pronunciation of

which is highly unpredictable, like proper names for example. In this context, it is difficult to use *a priori* knowledge, such as letter-to-sound rules in a reliable way.

In standard procedures like those presented in [7] and [8] for example, the speech utterance of the new word is aligned with speaker-independent allophone models. Effectively a decoding from utterance to allophone sequence is performed, with a bigram model on allophones functioning as the language model on words does in a normal utterance-to-text decoding; we will refer to this as the transition model. The resulting sequence of decoded allophones is mapped to a sequence of phones, yielding a baseform for the utterance.

The way to optimally combine the acoustic score given by the speaker-independent allophone models and the phonotactic score given by the transition model is an open issue as it is not known in advance which of the models can most reliably describe the acoustic evidence observed for each new word. For example, when the enrolled words are proper names, the reliability of the transition model is questionable since proper names do not follow strict phonotactic rules. Current techniques of automatic baseform generation do not take into consideration the relative degree of confidence that should be put in either modeling component.

B. Automatic generation of multiple pronunciations

This section summarizes our scheme to automatically derive pronunciations(also see [6] for more details). As in a typical speech-to-text decoding, a weighted combination of acoustic model and transition model log-probabilities is used to determine the best allophone sequence. However, it differs from the earlier approaches of [7] and [8] in that the speaker-independent allophone models and the transition model are assigned a weight of $(1 - \lambda)$ and λ respectively. Each value of λ defines a distinct likelihood objective function which reaches its maximum value for possibly distinct strings of allophones. Multiple baseforms are derived from a single speech utterance by varying the value of λ. All the distinct phonetic baseforms obtained from the speech utterance of a word by scanning a set of values of λ are added as pronunciation variants in the recognition lexicon.

The advantage of this approach is twofold. First, since we have to deduce the pronunciation of the enrolled words from just one or two speech examples, we may as well use multiple guesses to maximize the chance that one of them will be right. Second, since we do not know *a priori* if either the acoustic model or the transition model is more reliable, we avoid arbitrarily favoring either one of them by varying their relative weights when generating the guesses.

Each set of λ values results in a specific recognition lexicon, hence raising the question of how to select *a priori* the best performing lexicon. We can expect that accumulating multiple baseforms for each enrollment speech utterance will improve the recognition accuracy by allowing a broader modeling of the pronunciation of the new words. However it is well known that increasing the number of pronunciation variants increases the acoustic confusability in the recognition lexicon, which eventually hurts the accuracy. In our experiments, we noticed

for example that the baseforms obtained with λ equal to or more than 0.8 tended to look more and more alike, which we attributed to the prevailing influence of the transition model. As a result, cumulating baseforms with λ values higher than 0.8 was resulting in higher word error rates. In the following section, we report on experiments where lexicons are build for each interval $[\lambda 1; \lambda 2]$ with $\lambda 1$ in $\{0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7\}$ and $\lambda 2$ in $\{\lambda 1; ...; 0.7\}$. The selection of the most promising lexicon thus resumes to selecting the appropriate interval $[\lambda 1; \lambda 2]$.

C. Evaluation

For our experiments, we estimated the parameters of the bigram model of allophones from an aligned corpus of about 17,000 utterances, consisting primarily of names, addresses and digits We show results obtained with 2 different sets of enrolled words. To form the first set of enrolled words, we recorded 10 speakers each uttering twice 50 new words. To form the second set of enrolled words, we recorded 20 speakers each uttering only once 35 new words. In both sets, all enrollment utterances are recorded in a quiet environment. We then recorded test data for each of these two sets. In the test data corresponding to the first set, each of the same 10 speakers uttered the 50 new words in isolation 10 times and in short command sentences like "CALL ⟨name⟩", 10 times again. In the test data corresponding to the second set, each of the same 20 speakers uttered the 35 new words in short command sentences, this time in a car moving at the three speeds noted above.

Figure 6 plots the Word Error Rate as a function of the interval $[\lambda 1; \lambda 2]$ scanned to generate the multiple pronunciations. The points along the abscissa correspond to the intervals $[0.1; 0.1]$, $[0.1; 0.2]$, ... ,$[0.1; 0.7]$, ..., ending with the intervals $[0.6; 0.6]$, $[0.6; 0.7]$ and $[0.7; 0.7]$. The WER corresponding to a standard generation system ($\lambda 1 = \lambda 2 = 0.5$) are circled. The solid line and the dot line represent the WER averaged over all the speakers for the tests performed with the first enrollment set and the second enrollment set respectively. As can be seen, both curves show the same local patterns: the WER decreases along each portion of the x axis going from an interval $[\lambda 1; \lambda 1]$ to an interval $[\lambda 1; 0.7]$, which indicates how accumulating baseforms systematically improves the overall accuracy. Also, the general pattern of both WER curves is to increase towards the intervals starting with a $\lambda 1$ more than 0.5. The curves on Figure 6 tend to indicate that a close-to-optimal accuracy can be obtained by building a lexicon using an interval $[\lambda 1; 0.7]$, where $\lambda 1 \leq 0.3$. In our experiments, scanning the interval $[0.1; 0.7]$ yields relative WER improvement over a standard approach ranging from 20 to 40% depending on the testing condition.

Table V shows the WER with the test performed on the second set of enrolled words at each speed, by scanning the interval $[0.1; 0.7]$, for models BIC with variable alignments and MMI 3 described earlier. This same table includes two additional lines of results, both marked "(filtering)". Inspection of the automatically-derived baseforms showed that they contained implausible sequences of intermixed silence and consonantal baseforms, at those portions of the new word utterances corresponding to the start and finish of each new

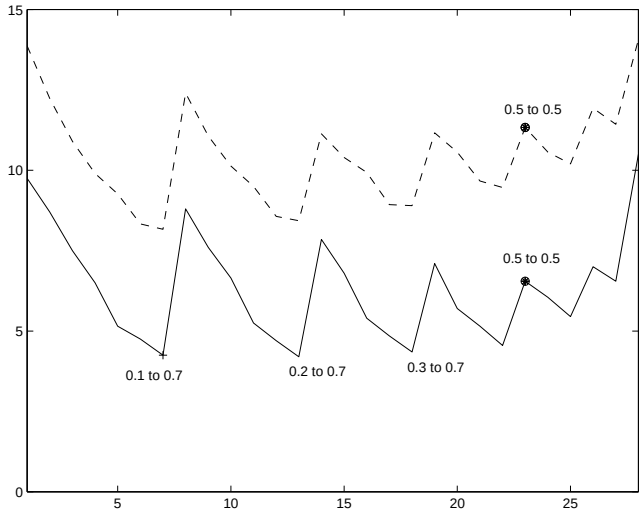


Fig. 6. WER as function of the scanned interval $[\lambda 1; \lambda 2]$ averaged over all speakers for the tests performed with the first enrollment set (solid line) and the second enrollment set (dot line).

Model	0 mph	30 mph	60 mph
MMI 3	9.5	9.2	12.2
BIC variable alignments	8.9	8.5	13.3
MMI 3 (filtering)	8.7	7.8	10.3
BIC var align (filtering)	7.6	7.8	10.1

TABLE V

WER (%) FOR DECODING OF AUTOMATICALLY GENERATED BASEFORMS.

word (thus the silence/speech and speech/silence transitions of each recording). To compensate for this noise we filtered the text of the automatically generated baseforms to remove such sequences. This yielded the performance improvements reported in the table.

VII. SUMMARY

This paper described a series of techniques to address robustness, recognition accuracy, system size, and computational resource issues in phonetically-based, low-resource, medium-vocabulary speech recognition systems. Speech recognition is already being investigated as a user interface for handheld computers [9], [10]; commercial systems cannot be far behind. Though microelectronic technology continues to advance in miniaturization and performance, we believe there will always be room at the bottom. For even as high-performance architectures migrate into handheld devices, the game remains afoot to make those devices smaller, sleeker, lighter, faster, easier to use—and hence ever more requiring accurate, low-resource speech recognition.

VIII. Acknowledgments

We thank Stanley Chen, Ellen Eide, P.S. Gopalakrishnan, Dimitri Kanevsky and Jan Sedivy for many useful discussions and contributions.

References

[1] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics,* **6**, pp. 461–464, 1978.

[2] S. S. Chen and R. A. Gopinath, "Model Selection in Acoustic Modeling," Proc. Eurospeech 99, Budapest, Hungary, September 1999.

[3] M. Mohri, "Finite-state transducers in language and speech processing," *Computational Linguistics*, 23(3), 1997.

[4] John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach. Morgan-Kaufmann Publishers, Palo Alto, CA, 1990.

[5] L.R. Bahl et al., "Performance of the IBM Large Vocabulary Continuous Speech Recognition System on the ARPA Wall Street Journal Task," ICASSP 1995, vol.1, pp 41-44.

[6] Sabine Deligne, Benoit Maison and Ramesh Gopinath, "Automatic generation and selection of multiple pronunciations for dynamic vocabularies," ICASSP 2001, Salt Lake City, UT, May 2001.

[7] R. C. Rose and E. Lleida, "Speech Recognition using Automatically Derived Baseforms," ICASSP 1997, pp 1271-1274.

[8] B. Ramabhadran, L.R. Bahl, P.V. DeSouza and M. Padmanabhan, "Acoustics-Only Based Automatic Phonetic Baseform Generation," ICASSP 1998.

[9] L. Comerford, D. Frank, P. S. Gopalakrishnan, R. Gopinath, J. Sedivy. "The IBM Personal Speech Assistant," ICASSP 2001, Salt Lake City, UT, May 2001.

[10] W. R. Hamburgen, D. A. Wallach, M. A. Viredaz, L. S. Brakmo, C. A. Waldspurger, J. F. Bartlett, T. Mann, K. I. Farkas, "Itsy: Stretching the Bounds of Mobile Computing," IEEE Computer, April 2001, pp 28–36.

[11] P.S. Gopalakrishnan, D. Kanevsky, A. Nádas and D. Nahamoo, "An inequality for rational functions with applications to some statistical estimation problems," IEEE Transactions on Information Theory, 37(1), January 1991.

[12] P.C. Woodland and D. Povey, "Large scale discriminative training for speech recognition," Proceedings of the Workshop on Automatic Speech Recognition, Paris, France, September 2000.

[13] Eric D. Sandness and I. Lee Hetherington, "Keyword-based discriminative training of acoustic models," Proceedings of ICSLP 2000, Beijing, PRC, October 2000.

[14] Y. Normandin, "Hidden Markov Models, Maximum Mutual Information Estimation and the Speech Recognition Problem", PhD Thesis, McGill University, Montreal, 1991.

[15] E. Weinstein, M. Feder and A.V. Oppenheim, "Multi-channel signal separation by decorrelation", IEEE Transactions on Speech and Audio Processing, vol. 1, num.4, October 1993.

[16] P. Gomez-Vilda, A. Alvarez, R. Martinez, V. Nieto and V. Rodellar, "A hybrid signal enhancement method for robust speech recognition", Proceedings of the Workshop on Robust Methods for Speech Recognition in Adverse Conditions, Tampere, Finland, 1999.

[17] A. Acero and R.M. Stern, "Environmental robustness in automatic speech recognition," Proceedings of ICASSP 90, 1990.

[18] M. Westphal and A. Waibel, "Model-combination-based acoustic mapping", Proceedings of ICASSP 01, 2001.

[19] S. Dharanipragada, R. Gopinath, B.D. Rao, "Techniques for capturing temporal variations in speech signals with fixed-rate processing," Proceedings of ICSLP98, 1998.