Low-Resource Speech Recognition of 500-Word Vocabularies

Sabine Deligne, Ellen Eide, Ramesh Gopinath, Dimitri Kanevsky
Benoit Maison, Peder Olsen, Harry Printz, Jan Sedivy

IBM Watson Research Center
Yorktown Heights, NY 10598 USA
printz@us.ibm.com

Abstract

We describe techniques for enhancing the accuracy, efficiency and features of a low-resource, medium-vocabulary, grammar-based speech recognition system. Among the issues and techniques we explore are front-end speech / silence detection to reduce computational workload, the use of the Bayesian information criterion (BIC) to build smaller and better acoustic models, the minimization of finite state grammars, the use of hybrid maximum likelihood and discriminative models, and the automatic generation of baseforms from single new-word utterances. We report WER figures throughout, as appropriate.

1. Introduction

In this paper we describe 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 processor offering approximately 50 DMIPS, augmented by perhaps 1 MB of DRAM.

This objective is both appealing in its promise and daunting in its technical challenges. Precisely because such systems are by highly portable, they are taken everywhere; thus noise robustness in adverse acoustic environments is an important issue. Moreover because such systems are intended for mass market appeal, the hardware must be low cost. Likewise because of the desire to appeal to the consumer, the system must not require a complicated enrollment procedure, yet must offer the user the ability to easily enter new words in the field. As a portable system, it is necessarily battery-powered; the batteries have to last an acceptably long time.

The plan of this paper is as follows. We begin with a general overview of the system architecture. We then discuss a series of techniques to address recognition accuracy, system size, and computational resource issues. Then we describe 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.

2. System Organization

Logically the system is divided into three primary modules: the front end, the labeler and the decoder. 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 operates at a 15 ms frame rate, computing 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.

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 hidden Markov models (HMMs). This computation yields the top 100 HMM states, to which likelihoods are assimilated based upon rank. The sequence of rank likelihoods is then forwarded to the decoder.

The decoder implements a synchronous Viterbi search over an active vocabulary of up to 500 words. 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 of 39-dimensional diagonal Gaussians. A key issue is whether or not the phone context is permitted to extend across word boundaries. We have investigated both approaches and settled on a system that uses within-word context only (that is, context does not extend over word boundaries). Except as noted in Section 5 all results in this paper are for such systems. We discuss this issue further below.

3. 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. 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.

4. Acoustic Model Size Reduction

The current acoustic model comprises 680 context-dependent observation densities (allophones), constructed from a total of just over 10,000 Gaussians. In a resource-constrained system, model size has significant economic and energetic 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 in 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 loglikelihood 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 loglikelihood) 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

$$BIC(\mathcal{X}, \mathcal{M}) = \log P(\mathcal{X} \mid \mathcal{M}) - \frac{\lambda M \log(N)}{2} \ . \qquad (1)$$

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 (1) 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, since $P(\mathcal{M}) \propto N^{-\lambda M/2}$.

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. Each phoneme in the model is subdivided into allophones by use of a decision tree, with each allophone modeled by a Gaussian mixture. Choosing n_g to maximize (1) 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 (1) 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 allophones 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 allophone for any given value of λ. Choosing a particular λ then determined the BIC-optimal model for each allophone; the collection of these allophone models then constitutes a complete acoustic model of a particular size. The resulting decoding accuracy for various model sizes is plotted in Figure 1.

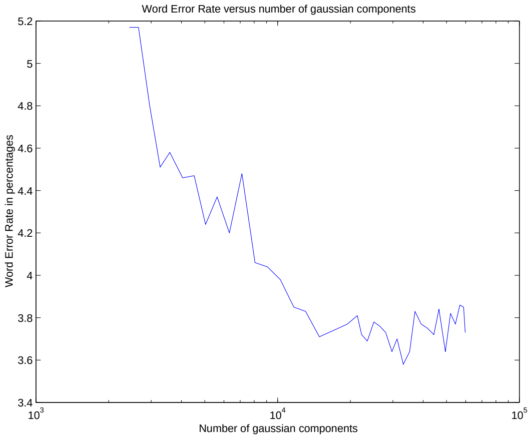

Figure 1: WER vs n_g for BIC with Fixed Alignments.

If desired these models may be further trained using variable alignments. Table 1 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.

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

Table 1: WER (%) for Systems Built with and without BIC.

5. Grammar Minimization

A finite-state grammar can be represented as a weighted finitestate 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 only we 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 2 (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 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

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 2: Sizes of Initial and Minimized HMM Graphs.

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 2. 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 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.

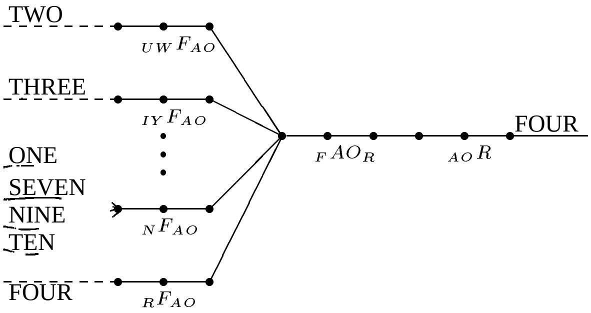

Figure 2: Subgraph of a Digits Grammar. This figure shows how intra-word context influences graph structure at the start of a word.

6. Hybrid ML/MMI Modeling

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.

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

Table 3: Performance of ML/MMI Hybrid Acoustic Model. Each column contains WER (%) and SER (%) respectively. See text for discussion.

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 (hereafter 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. We adopted the MMI training procedure described in [5], using the thresholding scheme proposed in [6].

The advantages of this 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 [7] 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 reestimated 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.

Our hybrid ML/MMI system has 89 phones (37 of which are digit phones), comprising 680 context-dependent allophones (triphone contexts tied by using a decision tree [8]), which are in turn modeled with 10508 Gaussians. This is in fact the baseline system of Table 1 above. Table 3 gives experimental results for three acoustic models, respectively baseline, MMI 1 (after one iteration of MMI training) and MMI 3 (after 3 iterations), for a speech recognition system operating in a car. The WER (word error rate) and SER (string error rate) performance of each model is given for a car moving at the velocities 0 mph (0 kph), 30 mph (50 kph) and 60 mph (100 kph) respectively, on each of four tasks: addresses (A), commands (C), digits (D) and radio control (R).

7. Acoustic Baseforms

An additional feature of our system is dynamic vocabulary expansion. This means that the user can add new words to the

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 4: WER (%) for Decoding of Automatically Generated Baseforms.

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.

The procedure used to automatically derive pronunciations is extensively described in [9]; we summarize it now. The speech utterance of the new word is aligned with the 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 estimated the parameters of this allophone bigram model from an aligned corpus of about 17,000 utterances, consisting primarily of names, addresses and digits; 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.

Our scheme differs from the earlier approaches of [10] and [11] as follows. Again as in a typical speech-to-text decoding, a weighted combination of acoustic model and transition model logprobs is used to determine the best allophone sequence. By varying the relative weights of the models, it is possible to derive multiple baseforms. 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.

As the weights are adjusted, all distinct baseforms obtained from the speech utterance of a new word are added to the recognition lexicon as pronunciation variants. In [9] we show that this multiple pronunciation scheme provides a relative decrease of the word error rate ranging from 20% to 40% over other techniques, depending on the test conditions.

To test the performance of this feature we first recorded 20 speakers each uttering 35 new words. Each word was uttered on time only, in a quiet environment. We then recorded test data for these same speakers, consisting of sentences like "CALL ⟨name⟩" for each added word, but this time recording the speakers in a car moving at the three speeds noted above. Each speaker uttered each name once at each speed. Table 4 shows the WER, for the test words only, 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 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.

8. Summary

We have explored techniques for improved accuracy, efficiency and appeal of phonetically-based low-resource, medium-vocabulary speech recognition systems. Such systems are already the basis of experimental designs [12]; 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 make their way into handheld devices, the game is always afoot to make one that is smaller, sleeker, lighter—and hence requiring speech recognition technology even more to make it easy and effective to use.

9. Acknowledgements

We thank Stanley Chen for his work on the Taconic code base, which was used to conduct some of the experiments reported in this paper.

10. References

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

[2] S. S. Chen, E. Eide, M. J. F. Gales, R. A. Gopinath, D. Kanevsky and P. Olsen, "Automatic Transcription of Broadcast News," *Speech Communication*, to appear.

[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] 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.

[6] 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.

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

[8] 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.

[9] 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.

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

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

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