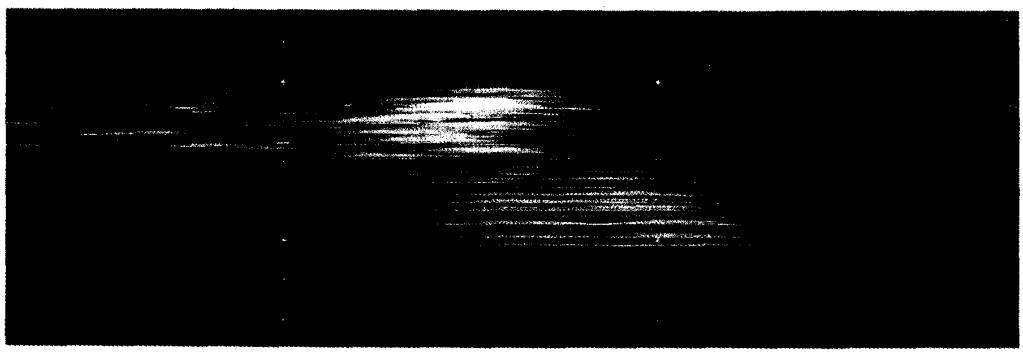
Continuous Speech Recognition Using Hidden Markov Models

Joseph Picone



Stochastic signal processing techniques have profoundly changed our perspective on speech processing. We have witnessed a progression from heuristic algorithms to detailed statistical approaches based on iterative analysis techniques. Markov modeling provides a mathematically rigorous approach to developing robust statistical signal models. Since the introduction of Markov models to speech processing in the middle 1970s, continuous speech recognition technology has come of age. Dramatic advances have been made in characterizing the temporal and spectral evolution of the speech signal. At the same time, our appreciation of the need to explain complex acoustic manifestations by integration of application constraints into low level signal processing has grown. In this paper, we review the use of Markov models in continuous speech recognition. Markov models are presented as a generalization of its predecessor technology, Dynamic Programming. A unified view is offered in which both linguistic decoding and acoustic matching are integrated into a single optimal network search framework.

0740-7467/90/0700-0026$1.00

Though automatic speech understanding by machine remains a distant goal in speech research, great strides have been made recently in the development of constrained, or application-specific, continuous speech recognition systems. Despite the fact that spoken language recognition still awaits more fundamental breakthroughs in linguistics, we are witnessing the emergence of structural methods [1] that promise to be the foundation upon which future speech understanding systems will be built. At the core of this new generation of technology are powerful statistical signal processing approaches that integrate detailed statistical characterizations of the acoustic signal with probabilistic models of application constraints.

In this review, we will restrict our scope to one particular class of statistical signal processing algorithms: first order Hidden Markov models (HMMs). Other variations [1-7] and generalizations [8-9] hold great promise towards extending the frontier of speech recognition technology, and share similar foundations in statistical estimation theory. We present Hidden Markov modeling as a generalization of its predecessor technology, Dynamic Programming (DP) [10, 11]. A unified view is offered in which both linguistic decoding and acoustic matching are integrated into a single optimal network search framework.

Advances in Recognition Architectures

The goal in continuous speech recognition is to provide a transcription of the incoming speech utterance, as depicted in Fig. 1. In this example, the speech signal has been decomposed into a sequence of phonetic units. This decomposition was the result of considering many possible explanations of the speech data given a model of all possible sentences that could be spoken, and choosing the best sequence based on some estimate of its likelihood. Information about the possible sentences that can be spoken, or language model, is represented using a hierarchy of representations that include general phonological principles and application specific information. (For example, doctors examine patients, not vice-versa.)

The output of the recognition system, in its simplest form, can be the sequence of words that was spoken. This task is often referred to as speech recognition (speech to text). More often, it is desired to have a system perform some useful function in response to a user's command, a task often referred to as speech understanding. In this case, it is frequently more useful to represent the information in a more abstract form [12], such as the logical form shown at the top of Fig. 1.

An example of a typical commercially available transcription system [13], *circa* 1984, is shown in Fig. 2. This system can be tersely described as a bottom-up [14] DP approach (also known as word lattice parsing). In the block labelled feature extractor, the digitized speech signal is converted into a sequence of feature vectors using a spectral analysis technique. Each feature vector corresponds to an interval of speech data, denoted a frame. The output of the feature extractor is processed by a DP based hypothesizer that incorporates an unconstrained endpoint DP algorithm [15]. Recognition hypotheses (words or any other user-defined recognition unit) are output asynchronously, whenever the match between the speech signal and a recognition model is reasonably good.

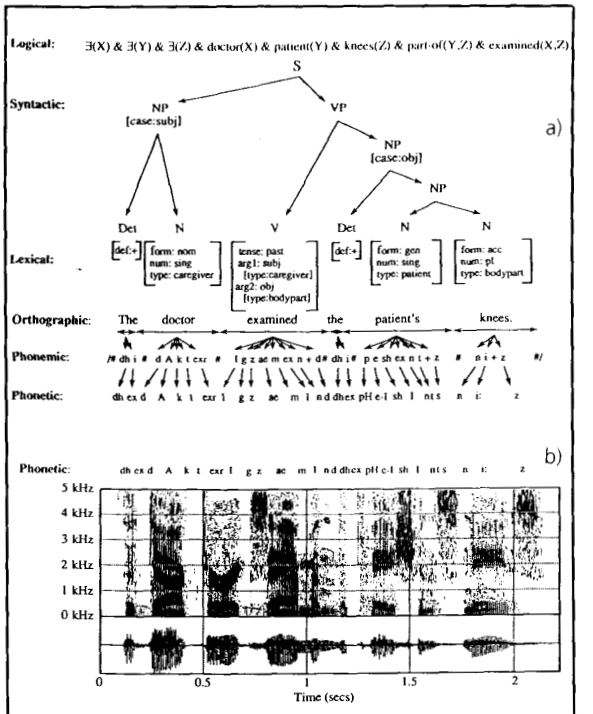


Fig. 1. With the emergence of structural methods in speech recognition, it is possible to tightly integrate application constraints into the recognition process, and to let knowledge of the application guide the potential choices of recognition units. Markov models provide a powerful paradigm for implementing such hierarchically organized systems.

a) A hierarchical labeling (or parse) of a speech signal depicting the influence application constraints can have on the speech recognition process. The verb "examine" in a medical context significantly limits the possible subject/object combinations. Differentiating between "patient's" and "patients'" requires additional context. (One might consider the subtlety of distinguishing the patient's knees from the patient's niece when "knees" is pronounced with a final unvoiced fricative).

b) A phonetic labeling of a speech signal. Previously, common approaches to phonetic recognition involved scanning the input utterance for probable locations of phones, and reducing these to a single transcription through the use of higher level knowledge (or application constraints).

The hypotheses generated by this low level DP recognizer are then postprocessed for the best "sentence" hypothesis by searching a finite state machine, or finite automation (FA) [14], for the lowest cost sequence of hypotheses. The system is described as a bottom-up system because information enters at the bottom (the speech signal), and exits in a distilled form at the top (the best sentence hypothesis output from the recognition system). No high level constraints encoded in the FA are exploited in the low-level recognition processing.

The main drawback to the DP scheme in Fig. 2 was that a sentence containing a particular word could not be recognized unless all words in the sentence were hypothesized by the low-level hypothesizer. The performance of the system, to a large

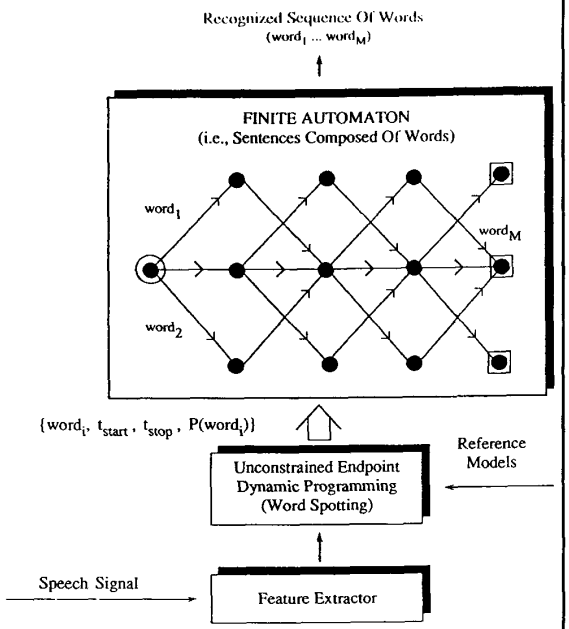


Fig. 2. A bottom-up DP speech recognition system that provides a pseudo-connected word recognition capability. A word hypothesis output by the unconstrained endpoint DP recognizer consists of a word index, the start and stop times of the hypothesis, and a measure of its similarity to the corresponding speech signal. These hypotheses are searched for the best grammatical sentence hypotheses using dynamic programming. The search procedure often incorporates many heuristics to deal with hypotheses that overlap in time, and to account for sections of the speech signal for which no hypotheses were generated. Reference models are often enrolled in an isolated word recognition mode, and updated on continuous speech.

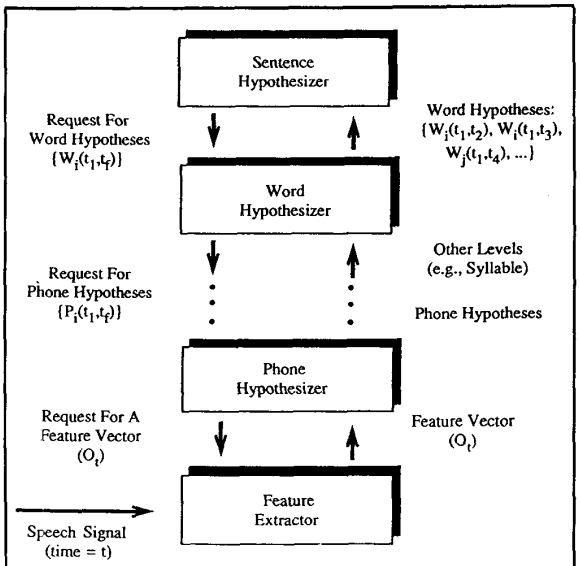


Fig. 3. A hierarchical architecture for continuous speech recognition based on top-down processing. Each level of the system can be represented as a Markov model, or other formalisms from the Chomsky hierarchy [11]. Alternately, the system can be compiled into a single level. Most recognizers today operate with two levels when using whole word recognition units, or three levels when using phone recognition units. The hierarchical architecture facilitates the training process in speech recognition.

extent, was limited by the performance of the low level hypothesizer. Configuring the low level hypothesizer to be less discriminating often made such systems computationally impractical and resulted in unacceptably high error rates.

An intuitively appealing approach to overcoming these deficiencies was to incorporate a top-down [14] information flow, in which requests for recognition hypotheses at a given point in time are initiated by the high level sentence processing. The low level recognition system initiates processing of a word model only after a request from the high level sentence processor has been received. The top-down paradigm is shown in Fig. 3. Recognition systems based on top-down search procedures (or similar techniques that exhaustively checked all possible combinations of recognition units) began supplanting bottom-up DP systems in the early 1980s [16, 17].

Top-down processing of FAs is one of many approaches to parsing [14]. In many applications, such as unconstrained speech understanding, top-down parsing is neither practical nor efficient. (For example, consider the number of nouns that can begin a sentence.) Today, however, we are finding an increasing number of applications, such as data entry tasks, in which small language models are useful. The top-down paradigm can be quite powerful for this class of applications.

The motivation for such hierarchical systems is based on an underlying model of the speech communication process in which the speech signal is a composite of hierarchically organized structures [18] that specify the permissible combinations of some fundamental set of units. The hope was that this type of hierarchy would bear some close resemblance to rule-governed linguistic theories. The exact constitution of each structure in this hierarchy, the topological constraints on the hierarchy, and the appropriate choice of fundamental units all remain issues of great debate within the linguistics and speech research communities.

Statistical Pattern Matching

Simultaneously with the development of improved network searching strategies, approaches to the problem of computing the similarity between a recognition model and a segment of a speech signal were being recast in a statistical signal processing framework using HMMs. One simple way to view the introduction of HMMs at the acoustic level is to consider HMMs as a generalization of the DP solution to the discrete time normalization problem [19], as depicted in Fig. 4.

In DP, time normalization and pattern matching are accomplished in a single discrete optimization procedure in which the incoming speech signal, decomposed into a sequence of feature vectors, is matched against precomputed reference vectors. Part of the system design includes development of heuristic functions [11] to impose global and local constraints on the time normalization process. Often these constraints impose a left to right structure on the mapping function, and effectively penalize de-

viations from a linear mapping. As in most optimization problems, there is no specific method for designing an optimal cost function (one that maximizes recognition performance) to balance these penalties.

In HMMs, the view of measuring acoustic similarity as a template matching problem is generalized to a problem of finding an optimal path through a recognition model.[1] The DP matching procedure in Fig. 4 is replaced with a computation of the probability of the speech data given a recognition model. There is no strict requirement to preserve the left to right structure of DP, though in practice, it is wise to do so. The major advantage in HMMs, from a time normalization perspective, is that a local constraint function can be reestimated, or optimized, by an iterative training procedure. Reestimation allows HMMs to assimilate the statistical characteristics of the training data, and to optimize performance on the training database.

Language Processing

Traditionally, the task of language processing has been separated from the task of acoustic matching. Language processing is often described in terms of automata theory and draws heavily on deterministic techniques commonly used in computer science. Acoustic matching, on the other hand, usually is described in statistical signal processing terms, such as HMMs. Only recently, with the introduction of stochastic language models in speech recognition [18] have the two tasks merged towards a common approach.

With the addition of statistical information, such as word probabilities, to finite automata, the descriptive power of a finite automation and an HMM are essentially equivalent. Often, differences are only cosmetic, in that language models have traditionally associated output symbols, such as words, with transitions between states, while HMMs have associated output symbols, feature vectors, with states. Throughout this paper, we will use the term HMM to represent both an acoustic model and a language model. The term symbol will mean both an orthographic unit, such as a word, and an acoustic unit, such as a feature vector, depending on which level in the recognition process is being discussed.

Isolated Word Recognition

One final introductory note regarding HMM based speech recognition is necessary. Previously, it has been convenient to distinguish recognition systems based on an ability to recognize isolated or continuous speech. Computational issues notwithstanding, it is advantageous in an HMM framework to consider the isolated word recognition problem using a continuous speech recognition framework. The basic advantage of this approach is that a heuristic utterance detection/segmentation algorithm is no longer needed: the recognizer determines the optimal start and stop times of an utterance.

The first network depicted in Fig. 5 shows an isolated word speech recognition task implemented using an HMM that allows an arbitrary duration of silence or nonspeech to precede or succeed a word. This is the analog in HMMs to DP with unconstrained endpoints [11, 15]. This network forces non-speech hypotheses to precede and succeed a word hypothesis. The word hypothesis is free to begin and end anywhere within the input utterance. By adding a transition from state s_2 to itself, the network in Fig. 5 accommodates a continuous speech recognition task in which any word can follow any other word, with arbitrary durations of non-speech occurring between any two words.

This example illustrates the flexibility of the hierarchical HMM approach. Achieving this level of flexibility was previously very difficult in a DP system. This poses a radical departure from the old bottom-up DP systems for two reasons. First, each recogni-

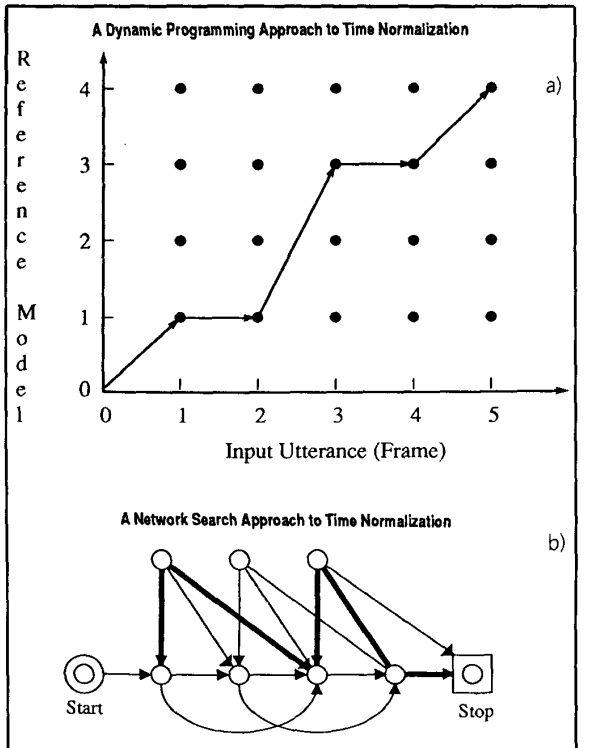


Fig. 4. An illustration of the similarity between Dynamic Programming and Hidden Markov model based time normalization. The HMM approach is a generalization of the traditional DP approaches for two reasons: 1) Time normalization is often less constrained, and 2) The properties of the network are statistically optimized.

a) In fixed endpoint dynamic programming, the optimal path is constrained to map frame 0 of the reference model to frame 0 of the input utterance, and to map frame 5 of the reference model to frame 5 of the input utterance. In unconstrained endpoint dynamic programming, the optimal path is not required to start at frame 0 of the input utterance.

b) The fixed endpoint dynamic programming problem of Fig. 4a can be formulated as a network search problem. The topology of the network can be arbitrarily complex. States in the network correspond to frames in the reference model of Fig. 4a. Each frame of speech data forces a transition between states. The darkened path represents the result of an optimal search computation to find the best match between the model and the speech data.

[1] Strictly speaking, this statement assumes use of the Viterbi algorithm (described in Section II) to compute the best state sequence, rather than the forward-backward procedure (described in Section III), which computes the probability of all possible state sequences.

tion hypothesis must now account for the entire speech signal. Second, rejection of background noise and out of vocabulary utterances will largely rest on an ability to characterize a class of extraneous signals using explicit models. In the example in Fig. 5, silence, channel noise, acoustic noise, and out of vocabulary speech will all be lumped into the non-speech model.

Thus, the speech recognition problem, reduces to two basic tasks: (1) searching for an optimal path through a hierarchy of HMMs, and (2) computing some measure of similarity between the acoustic signal (a feature vector) and a stored model (reference vector). In the next section, we discuss the fundamentals of Viterbi beam search, the dominant search algorithm used today in speech recognition. In Section III, we discuss approaches to estimating the probabilities associated with an HMM model. In Section IV, we present the HMM supervised training paradigm. Finally, in Section V, we review several examples of successful HMM based speech recognition systems.

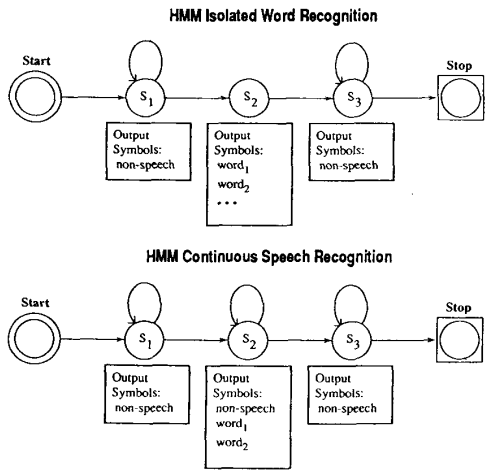


Fig. 5. Common HMM structures that are used for isolated word recognition and continuous speech recognition. In systems that require one frame of speech data to produce a transition, background noise and out of vocabulary utterances will be modelled by the non-speech model.

VITERBI BEAM SEARCH

Let us formally define a discrete observation HMM as:

$$G = (S, V, A, B, \pi), \tag{1}$$

where $S = \{s_1, s_2, \ldots, s_N\}$ are the N individual states, $V = \{v_1, v_2, \ldots, v_M\}$ are the M output symbols, A is an $N \times N$ matrix containing the state transition probabilities:

$$A = \{a_{ij}\}, \qquad a_{ij} = P(s_j \text{ at } t+1 / s_i \text{ at } t), \tag{2}$$

B contains the observation symbol probability distributions:

$$B = \{b_j(k)\}, \qquad b_j(k) = P(v_k \text{ at } t / s_j \text{ at } t), \tag{3}$$

and π are the initial state probabilities:

$$\pi = \{\pi_i\}, \qquad \pi_i = P(s_i \text{ at } t = 1). \tag{4}$$

The variable t denotes discrete time. Note that B is often implemented as an $N \times M$ matrix in a discrete observation HMM.

There is one additional useful parameter that can be added to the HMM of Eq. 1: a state duration distribution [20, 21] that controls the amount of time spent in a given state. State durations are primarily useful in the development of recognition models, and have not been used extensively in language modeling. An alternate way of implementing a state duration model is to explicitly embed HMM topologies for each state in the original HMM model that models the desired duration distribution for the original state.[2] Thus, for simplicity and uniformity, we will not consider state durations as an explicit parameter in an HMM.

In continuous speech recognition, HMMs are used in a generative mode to begin hypothesis processing. At initialization (often considered the beginning of an utterance, or speech data file), all possible start symbols at the highest level are enumerated as separate hypotheses. The general objective of the search algorithm is to explore each of these hypotheses in some orderly fashion until the "best" explanation of the speech data is found. The details of this procedure depends heavily on the particular type of search algorithm used.

The Viterbi Algorithm

There is a vast amount of literature on the problem of optimal search techniques [21-25]. To exhaustively search a hierarchy of HMMs for an optimum solution, even for the simple HMMs shown in Fig. 5, is impractical. For example, an exhaustive search solution of a simple digit recognition task (any digit can follow any other digit) results in a number of hypotheses proportional to M^T, where M is the number of words in the vocabulary and T is the total number of frames of speech data. Fast search techniques produce functionally equivalent solutions, yet only search a fixed number of hypotheses versus time (or use a fixed amount of memory). Fast search techniques can be as much as two orders of magnitude more efficient in processing and memory requirements for very simple recognition problems. For realistic problems, exhaustive search techniques are simply not computable (consider $M = 1000$ and $T = 100$).

The Viterbi algorithm [22] is an efficient algorithm for finding an optimal solution. It is based on the Principle of Optimality [10], and has been used extensively in DP based speech recognition. It imposes the restriction that the cost, or probability, of any path leading to a given state can be computed recursively as the sum of the cost at the previous state, plus some incremental cost in making a transition from the previous state to the current state. This constraint integrates nicely with the temporal constraints imposed by a Markov model.

The Viterbi algorithm, used within a single Hidden Markov model [26, 27], can be summarized as follows:

Initialization ($t = 1$):

$$\delta_1(i) = \pi_i b_i(O_1), \qquad 1 \le i \le N, \tag{5}$$

$$\Psi_1(i) = 0, \tag{6}$$

Recursion:

$$\text{For } t \ge 2, \qquad 1 \le j \le N,$$

$$\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \tag{7}$$

$$\Psi_t(j) = \operatorname*{argmax}_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}], \tag{8}$$

[2] An example of an HMM topology with a state duration model implemented explicitly by an embedded HMM topology is given in Fig. 9(c). Fig. 9 is discussed in Section V.

Termination:

$$\delta^* = \max_{1 \le i \le N} [\delta_T(i)], \tag{9}$$

$$i_T^* = \operatorname{argmax}_{1 \le i \le N} [\delta_T(i)], \tag{10}$$

Backtracking:

$$\text{For } t = T-1, T-2, \ldots, 1,$$

$$i_t^* = \Psi_{t+1}(i_{t+1}^*), \tag{11}$$

where O_t denotes the discrete symbol observed at time t. Often, rather than accumulating products of terms, probability computations are carried out in the log domain.

The importance of transition probabilities and observation probabilities can be adjusted in Eqs. 7 and 8 by weighting each probability. For example, a_{ij} can be replaced by $(a_{ij})^\alpha$. α is often referred to as the language model weight, since α controls the contribution the transition probabilities of the language model have in the overall sentence hypothesis probability. (α is also useful in acoustic modeling as a means of adjusting the contributions of transition probabilities in acoustic model scoring).

The key calculation in the Viterbi algorithm occurs in Eq. 7. The only path propagated is the most probable path selected from all possible paths that can make a transition to the current state at time t. We introduce this process of eliminating alternate choices as our first method of limiting the search space, or pruning. The Viterbi algorithm is optimal in the sense that none of the discarded paths can ever be more probable than the path that is propagated. Unfortunately, the reduction in the search space achieved by the Viterbi algorithm still does not make most continuous speech recognition tasks tractable.

Viterbi Beam Search

A common goal with most fast search algorithms is reduction of the search space over which all acceptable hypotheses must be evaluated by establishing some measure of goodness of a hypothesis. This is generally denoted as a beam search problem (from the analogy that objects that fall outside of a beam of light fall into darkness). Hypotheses that fail the goodness test are discarded, or pruned. Beam search approaches are generally sub-optimal: occasionally a hypothesis that might prove to be the best global hypothesis is discarded at some prior point in the search. Fortunately, in speech recognition, sub-optimal solutions tend to produce useful (and accurate) results.

There are three characteristics of the speech recognition task that make the beam search problem non-trivial. First, the probability of all hypotheses always decreases with time, since the probability of a hypothesis is the product of probabilities that are normally (significantly) less than one. Thus, direct comparisons of hypotheses that account for different amounts of speech data are difficult. Second, uncertainty in the observation of any symbol mandates entertaining many alternate hypotheses. Acoustic matching is not refined to the point where hard decisions can be made at a low level. Finally, since the true start and stop times of a hypothesis are unknown, we must ntertain many hypotheses of the same symbol that differ primarily in start and stop times.

Many beam search approaches differ in the details in which the search is carried out. Some algorithms are depth-first [14]: the most probable path is repeatedly expanded until a stop node is encountered, thereby establishing some bound on the cost of an acceptable solution. Other approaches are breadth-first [14]: all hypotheses are expanded simultaneously. It is the latter approach that we currently find most interesting for speech recognition.

Viterbi beam search incorporates a breadth-first search strategy into the Viterbi algorithm in a time synchronous fashion. The basic principle is that at any point in time, we need only keep all hypotheses whose probabilities fall within some threshold of the best global hypothesis. Viterbi beam search can be summarized as follows:

```
Initialization:
  Activate Initial Hypotheses (Top Level)
Recursion:
  For all speech data
    For all levels
      For all active states
        Extend each active hypothesis by one frame
        Perform the Viterbi algorithm
        Pruning
      End active states
    End levels
  End speech data
Termination:
  Choose most probable hypothesis
```

Details of various implementations of Viterbi beam search can be found in Lee and Rabiner [21] and Ney *et al.* [28].

There are a number of significant bookkeeping issues that go into managing the complexity of this search process. The most important considerations include ensuring evaluation of each recognition model at most only once per frame, and visiting only those states that are active at any level. For large language models with highly constrained grammars, processing by iterating over active hypotheses is useful, since the number of active hypotheses may be small compared to the number of states, and all states at a level may not be active. On the other hand, for language models with few constraints, all states are likely to be active, and state oriented processing is efficient.

This process of extending each active hypothesis requires registering new requests for symbol hypotheses from the next lower level, and coordinating the return of these hypotheses (with probabilities) at a later point in time. The Viterbi algorithm step considers only hypotheses that have no pending requests for symbol explanations at a lower level (a portion of the active hypotheses will be waiting for a symbol probability to be returned by the lower level). A hypothesis that has no outstanding requests for a symbol probability can make a transition between states in an HMM, and hence can be processed by the Viterbi algorithm, since only the best hypothesis at any state at point in time must be retained.

Pruning, the process of discarding unpromising hypotheses, is accomplished by discarding all hypotheses whose probability falls below a certain threshold of the global best hypothesis [29]. A simple decision rule to implement pruning is:

$$P_H(t) > P_{H^*}(t) - \delta(t), \tag{12}$$

where P_H is the accumulated probability of the current hypothesis being tested, P_{H^*} is the probability of the best hypothesis cur-

rently active, and $\delta(t)$ is a threshold, or beam width. Because the search is time synchronous, all hypotheses explain the same amount of speech data, and hence need not be normalized before pruning.

The amounts of memory and processing required in Viterbi beam search are proportional to the number of active hypothesis that fall within the beam. This is loosely correlated with $\delta(t)$, the independent variable in the search algorithm. Processing time is often exponentially proportional to $\delta(t)$. Pruning frees memory that can be reused in future processing. If the search is highly ambiguous, thereby generating a large number of competing hypotheses, longer processing times and more memory will be required. Recognition accuracy is also typically related to the ambiguity of the search: longer processing times are often symptoms of poor performance.

ESTIMATING PROBABILITIES IN HMMS

The input to the hierarchical search procedure described in Section II is the probability of observing a segment of the speech data given an HMM. The first method introduced to allow efficient computations of this probability, known as the forward-backward procedure [30], followed in the true spirit of a doubly stochastic system. Since states of the hidden stochastic processes cannot be directly observed, the single path that actually produced the observation sequence is essentially unknown. There are typically many ways in which a single observation sequence can be produced in an HMM. The Viterbi algorithm, presented in the previous section, based a symbol probability on the probability of the single most probable path through the model.

The forward-backward procedure is based on an approach in which the symbol probabilities are estimated as the sum of the probabilities of all paths that could have produced the observation sequence. The symbol probability computation using the forward-backward procedure can be summarized as follows [26, 27]:

$$P\{O/G\} = \sum_{i=1}^{N} \alpha_T(i), \tag{13}$$

where

$$\text{for } 1 \le i \le N,$$

$$\alpha_1(i) = \pi_i b_i(O_1), \tag{14}$$

$$\text{for } t = 1, 2, \ldots, T-1, \quad \text{and } 1 \le j \le N$$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \tag{15}$$

and,

$$\text{for } 1 \le i \le N,$$

$$\beta_T(i) = 1, \tag{16}$$

$$\text{for } t = T-1, T-2, \ldots, 1, \quad \text{and } 1 \le j \le N,$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j). \tag{17}$$

Perhaps the most significant aspect of HMMs in relation to speech processing is the existence of an iterative training procedure in which the parameters of a Hidden Markov model can be adjusted to better represent the statistics of a training database (a representational technique). One such approach, the Baum-Welch method [30], is based on the maximum likelihood principle. A new model is computed that is guaranteed to improve the probability of the observation sequence given the model. The Baum-Welch procedure can be summarized as follows:

$$\hat{\pi}_i = \text{expected no. of times in } s_i \text{ at } t = 0,$$

$$= \alpha_1(i)\beta_1(i), \tag{18}$$

$$\hat{a}_{ij} = \frac{\text{expected no. of transitions from } s_i \text{ to } s_j}{\text{expected no. of transitions from } s_i}$$

$$= \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)}, \tag{19}$$

$$b_j(k) = \frac{\text{expected no. of times in } s_j \text{ and observed symbol } v_k}{\text{expected no. of times in } s_j},$$

$$\hat{b}_j(k) = \frac{\sum_{\substack{t=1 \\ O_t = v_k}}^{T-1} \alpha_t(i)\beta_t(i)}{\sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)}. \tag{20}$$

The Baum-Welch reestimation procedure is based on the intuitive notion that a new estimate of a transition probability can be based on the expected number of transitions from state i to state j, divided by the expected number of transitions out of state i. Similarly, the new output symbol probability for the kth symbol at state i is the expected number of times a symbol is output from the state divided by the expected number of times of being in the state. We use the term expected because these statistics are usually averaged over large amounts of data, and because the actual state transitions and output events are hidden.

The Baum-Welch reestimation procedure can be replaced with a much simpler Viterbi procedure based on the Viterbi algorithm [31]. Rather than compute the expectations of events, the actual counts are accumulated at each state based on the Viterbi best path calculation. For instance, a transition probability is reestimated by merely counting the number of times the transition is used and dividing it by the number of times the source state for the transition is used. This requires maintaining counters to track each transition and each output symbol during training.

The reestimation equations for the Viterbi algorithm are:

$$\hat{a}_{ij} = \frac{\text{no. of transitions from } s_i \text{ to } s_j}{\text{no. of transitions from } s_i}, \tag{21}$$

$$\hat{b}_j(k) = \frac{\text{no. of times in } s_j \text{ and observed symbol } v_k}{\text{no. of times in } s_j}. \tag{22}$$

The Viterbi algorithm is a popular alternative to the forward-backward procedure for two reasons. First, it is computationally more efficient than the forward-backward procedure, yet, in practice, it gives comparable recognition performance. Second, the Viterbi algorithm is easily extended to more general approaches to language modeling that depart from the finite automation structure [1, 31].

We have deferred perhaps the most important issue in speech processing for last. How do we actually compute observation probabilities given a speech signal?

Nonparametric Acoustic Measurements

HMMs were first introduced in speech recognition in the discrete observation form. A straightforward way to exploit HMMs was to build upon a feature extraction technique, such as Vector Quantization (VQ) [32], in which the speech signal is converted to a sequence of feature vectors drawn from a discrete distribution. VQ is a simple nearest neighbor classification technique in which a measured feature vector is quantized into one of a set of Q vectors using a minimum distortion criterion. The primary advantage of VQ from a statistical signal processing viewpoint is that complex vector spaces can be modeled with arbitrary precision by simply designing a sufficiently large codebook. Hence, VQ is referred to as a nonparametric modeling technique.

There are many approaches to feature analysis used today in speech recognition. Comparative analyses of different techniques can be found in [33, 34]. Most approaches share three common characteristics: LPC-derived features are used for computational efficiency; static measures of the spectrum are combined with short-term time differences that capture dynamic aspects of the spectrum; and multiple energy measures are included, such as absolute energy, normalized energy, and/or differenced energy. The most common front-ends for continuous speech recognition incorporate [35, 36] approximately 12 cepstral coefficients, the time-derivatives of these 12 cepstral coefficients, log energy and differential energy.

A codebook for a vector quantizer can be constructed from a set of feature vectors (considered the training sequence) by use of a hierarchical clustering algorithm. There are generally two approaches used today: the K-MEANS [37] or the Linde-Buzo-Gray algorithm [38]. Both of these algorithms generating a reduced dimensionality space by replacing groups of similar codewords (or clusters) in the training database with a single codeword that represents the group centroid. VQ codebooks ranging from 32 vectors for small vocabulary tasks (such as digit recognition) to 256 vectors for large vocabulary systems (such as phone-based recognition) are typically used.

A VQ system is conceptually simple in that the feature vector extracted from the speech signal is assigned a codeword in the vector quantization process by choosing the codeword producing minimum distortion. The details of the distortion measure often depend on the particular feature set used. However, in most cases, a weighted Euclidean distance measure is used. We defer the details of the distortion computation until the next section, in which we discuss stastically optimal techniques to compute distortion.

Parametric Acoustic Measurements

An alternate method to compute observation probabilities in an HMM is to compute an observation probability directly from the feature set, and avoid accumulating the distortion that might occur in the VQ process. The most common approach to doing this is the Continuous Density HMM (CDHMM) [39]. In CDHMM, we associate a multivariate Gaussian distribution with each state in the HMM:

$$b_j(\mathbf{O}) = \aleph[\mathbf{O}, \mathbf{u}_j, \mathbf{U}_j], \tag{23}$$

where $\aleph$ represents a Gaussian distribution whose mean is $\mathbf{u}$ and whose covariance is $\mathbf{U}$. $\mathbf{O}_t$ is a feature vector measured from the speech signal.

In the CDHMM model, we assume that the feature vector distribution encountered at a given state can be modeled by a Gaussian distribution whose underlying mean represents the "true value" of the feature vector at that state. A more general approach is to associate with each state a weighted sum, or mixture of Gaussian distributions [27]. Mixture distributions are capable of modeling arbitrarily complex distributions, similar to the VQ approach in the discrete case.

The output symbol probability distribution in CDHMM is reestimated in a similar manner to the discrete case using the forward-backward procedure:

$$\hat{\mathbf{u}}_j = \frac{\sum_{t=1}^{T-1} \alpha_t(j)\beta_t(j)\mathbf{O}_t}{\sum_{t=1}^{T-1} \alpha_t(j)\beta_t(j)}, \tag{24}$$

$$\hat{\mathbf{U}}_j = \frac{\sum_{t=1}^{T-1} \alpha_t(j)\beta_t(j)[(\mathbf{O}_t - \mathbf{u}_j)(\mathbf{O}_t - \mathbf{u}_j)^T]}{\sum_{t=1}^{T-1} \alpha_t(j)\beta_t(j)}, \tag{25}$$

In the case of a Viterbi approach, the mean of the distribution is simply reestimated by averaging all observation vectors that were associated with a given state during training. Similarly, the covariance matrix, in the Viterbi approach, is reestimated by computing the covariance for all vectors associated with a state during training. To summarize:

$$\hat{\mathbf{u}}_j = (1/N_j) \sum_{i=1}^{N_j} \mathbf{O}_i, \tag{26}$$

and,

$$\hat{\mathbf{U}}_j = (1/N_j) \sum_{i=1}^{N_j} (\mathbf{O}_i - \mathbf{u}_j)(\mathbf{O}_i - \mathbf{u}_j)^T, \tag{27}$$

where $\mathbf{O}_i$ is the ith observation vector associated with state j and N_j is the number of observation vectors associated with state j.

The More Things Change...

It is at this point that we have almost come full circle: that is, the CDHMM, with a single distribution per state, and a single output symbol per state, resembles a DP system with frame specific features [34]. If we process log probabilities, Eq. 23 simplifies to:

$$\log(b_j(\mathbf{O})) = (1/2)(\mathbf{O} - \mathbf{u}_j)\mathbf{U}_j^{-1}(\mathbf{O} - \mathbf{u}_j)^T - (1/2)\log(2\pi^P|\mathbf{U}_j|), \tag{28}$$

where $\mathbf{P}$ is the number of dimensions in the observation vector. The distance measure in Eq. 28 is equivalent to computing a Euclidean distance of a transformed observation vector:

$$\tilde{\mathbf{O}} = \Lambda^{-1/2}\Phi\mathbf{O}, \tag{29}$$

where Λ is a diagonal matrix of eigenvalues of $\mathbf{U}$, and Φ is a unitary matrix of eigenvectors of $\mathbf{U}$.

The transformation of Eq. 29 is known as a whitening transformation [40]. It performs a principal components analysis of $\mathbf{O}$, retaining all dimensions. Additional discrimination can be achieved by discarding the dimensions with the largest eigenvectors (dimensions of least discrimination). This procedure has been successfully employed in both DP [34] and HMM [41].

Over the years, much work as been done on appropriate ways to weight features to measure distortion in a more perceptually meaningful way. The HMM paradigm provides an explicit mechanism to compute optimal feature transformations on a per state basis. Eq. 28 effectively computes a Euclidean distance (actually known as the Mahalanobis distance) of two feature vectors after the input feature vector has been transformed into an orthogonal space in which each dimension has equal weight in the distance measure. This transformation decorrelates elements of the feature vector and equalizes the variances of each dimension. This is extremely important when mixing heterogeneous parameters in a feature vector (for example, mixing cepstral coefficients, differential cepstral coefficients, and energy).

There are two useful simplifications of CDHMMs. First, if the correlations between elements of a feature vector are small for the off-diagonal terms of the covariance matrix, as is often the case with cepstral coefficients, the covariance matrix of Eq. 23 can be approximated as a diagonal matrix of variances. This is known as variance-weighting, and is similar to several common weighted cepstral distortion measures [36].

Second, if the structure of the correlation matrix is similar from state to state, each covariance matrix can be replaced by a single covariance matrix representing the average of all covariance matrices. The distortion measure, in this case, resembles a traditional weighted Euclidean distance measure, similar to those used in discrete HMMs with a VQ front-end, and to those used in previous DP systems.

It is tempting to believe that the additional degrees of freedom allocated on a per state basis in CDHMM would result in significant improvements over discrete HMMs. In instances where large amounts of training data exist, and the vocabulary of the recognition task is small in size, this seems to be the case [41, 42]. For large vocabulary recognition tasks, however, this is still an open issue.

The CDHMM significantly increases the number of free variables that must be estimated, often by two orders of magnitude in an HMM system. Frequently, training databases are not large enough to support robust estimation of as many parameters as can be encountered in a CDHMM system. Further, whether all these degrees of freedom are true dimensions of discrimination in a large speech recognition task is an open question. Often, if improperly estimated, additional degrees of freedom can add significant amount of noise probability calculations, and degrade performance.

TRAINING SPEECH RECOGNITION SYSTEMS

The training procedure for an HMM-based speech recognition system is a three step process. In the first step, denoted seed model generation, an initial set of prototype models must be generated. The second step, reestimation, uses the maximum likelihood based techniques described in the preceding sections, to reestimate model parameters. Recently a third phase [43] in the training process has been introduced which seeks to improve recognition performance by enhancing the discrimination power of the reference models. The three-step training procedure is summarized in Fig. 6.

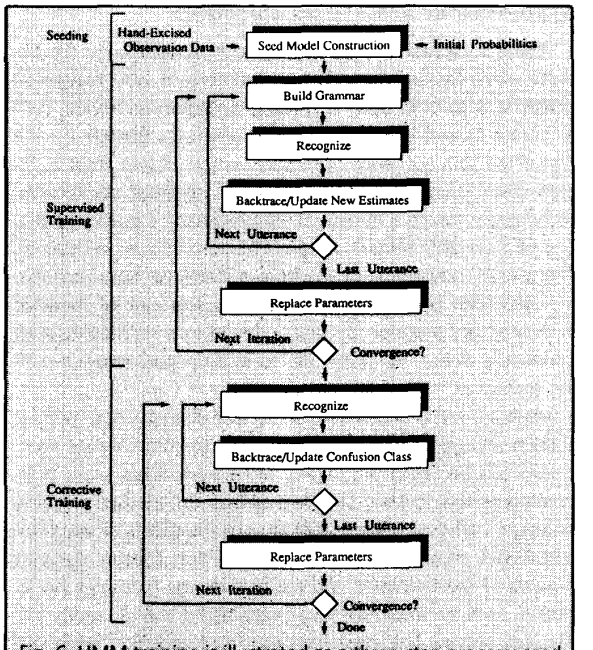


Fig. 6. HMM training is illustrated as a three-step process: seed model generation, supervised training, and corrective training. In the supervised mode, the system is forced to recognize an utterance using a reduced language model that allows only plausible pronunciations. *A priori* segmentation of the utterance is not required. Supervised training is a powerful paradigm because it forces the recognition models to capture contextual variations presented in the training database.

Seed Model Generation

The iterative training process, at a very basic level, is a nonlinear optimization technique. As with all such techniques, sensitivity to initial conditions is a concern: more so in CDHMM than in discrete HMM. Put very simply, good recognition performance is obtained by using good HMM reference models, which, in turn, are generated by choosing good seed models. HMMs leave a large number of free variables under the control of system designer. Some of these choices are analogous to similar issues in DP, and our intuitive sense of the correct procedure is well-developed (for instance, how many models should be used per lexical item?). Several key system parameters, such as the reference model topology, are not altered in the reestimation process, and hence, must be judiciously chosen before training.

One general observation about the development of good seed models is that model development is often an iterative process. Models using simple parameter sets are trained, and then successively refined and extended by bootstrapping from the models of a previous stage of the procedure. For instance, a CDHMM system using a mean and covariance per state will not be generated from scratch, but often built from the result of a CDHMM system using a single covariance matrix for all states. In general, we seek to minimize the number of free variables added to the system at any point in the training process.

The number of states in a model is often chosen to be proportional to the number of distinct acoustic events in the recog-

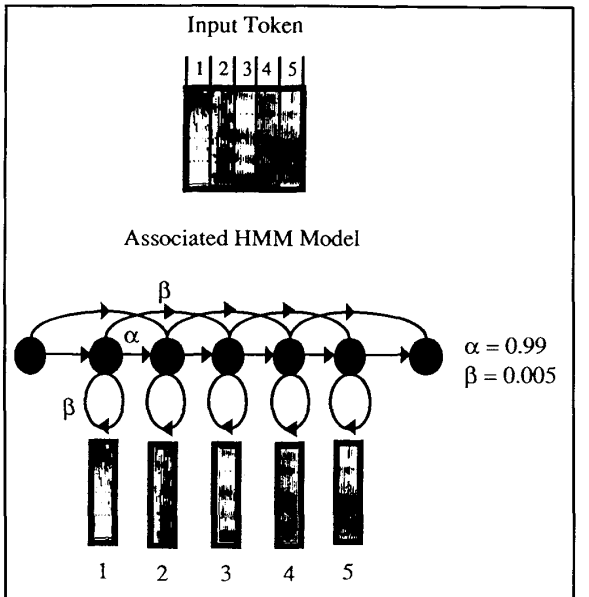


Fig. 7. Initialization of a CDHMM is often accomplished by hand-excising a representative token and creating one state for each frame in the token. Transition probabilities are initialized to favor a path through the model consisting entirely of progress transitions. The representative token is typically selected based on its duration (for example, in the model above, the number of states should be equal to the average duration in frames of the recognition unit).

nition unit (number of phones, for instance). At one end of the spectrum are phone-based HMM systems that use as few as 5 states per phone [35], while the other extreme is CDHMMs that use a number of states proportional to the average duration (in frames) of a unit [44]. The latter process is summarized in Fig. 7.

The transition model probabilities are typically initialized to reflect an equiprobable distribution (another approach is to initially favor the most intuitively appealling path through the model). Often, the initial observation means are hand-excised from a nominal pronunciation of the recognition unit. Clustering procedures, analogous to those used in DP, can be used to generate seed values that more closely reflect the state observation means [44, 45].

Supervised Training

Perhaps the most significant advancement introduced to speech recognition in HMM, supervised training, is not solely restricted to HMM. Simultaneously with the development of HMM and hierarchically organized recognition systems, training recognition units in context have become the dominant training technique for continuous speech recognition. Though the notions of HMM and supervised training are almost indistinguishable in the literature, it is important to point out that supervised learning, a notion familiar to other areas such as neural networks, is an extremely powerful formalism that applies equally well to DP and HMM.

The supervised training process, summarized in Fig. 6, is very simple. For each input utterance, the recognizer is constrained only to recognize the input utterance. The constrained grammar can be derived by reducing the application language model to a smaller model that generates only the orthographic transcription of the utterance. Next, the normal recognition process is executed for this constrained grammar. HMM parameters are then reestimated based on the recognition results.

For instance, for the utterance of Fig. 1, a grammar would be constructed that only allows the sentence "The doctor examined the patient's knees" to be recognized (along with an arbitrary amount of silence preceding and following the utterance). This is summarized in Fig. 8(a).

Note that, although the recognizer is constrained to recognize a particular transcription, the actual speech data need not be marked. This is an important practical consideration for two reasons. First, transcription of a database is a time-consuming process, and hence, there are very few large transcribed databases. Second, supervised training lets the recognizer capture context effects and permits reestimation of the recognition units to optimize recognition performance on the training database. Thus, the recognizer decides for itself what the optimal acoustic representation of a unit might be, rather than being heavily constrained by *a priori* knowledge based on human intervention. (This is, of course, both good and bad, depending on your point of view).

Often an utterance will be transcribed in terms of words, but not transcribed in terms of the actual lower-level recognition units. In this case, a partial supervision strategy can be used. In a hierarchical phonetic recognition system, for instance, all possible phonetic transcriptions for each word in the word-level orthographic transcription can be generated. Recognition of the utterance can then be performed using this less constrained

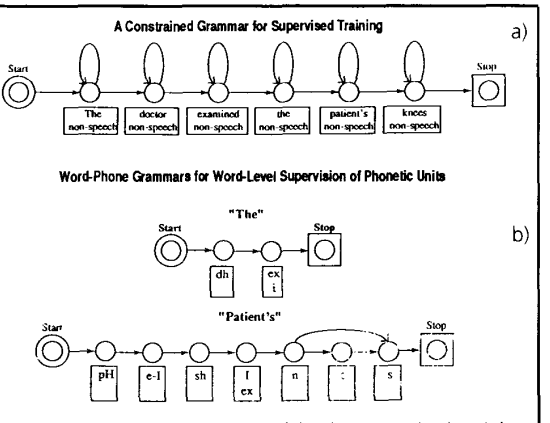


Fig. 8. The Build Grammar model in the supervised training paradigm of Fig. 6 is responsible for constraining the recognition task during training.

a) A constrained grammar that generates only the orthographic transcription of the input utterance. This grammar allows arbitrary amounts of non-speech to precede or succeed an utterance. Arbitrary amounts of non-speech are also allowed between each word. This permits robust reestimation of models on continuous speech utterances with *a priori* segmentation of the data.

b) In a hierarchical system, phone units can be trained using word-level supervision. Several phonetic transcriptions for each word in the lexicon are allowed during the recognition phase of training.

grammar, and HMM parameters reestimated accordingly. A less constrained grammar for the utterance of Fig. 1 that describes words in terms of phones is shown in Fig. 8(b).

In the supervised training procedure, reestimated parameters are substituted only at the end of an iteration through the entire training database, rather than after each utterance. The former situation is a block-oriented reestimation procedure. The latter has three interesting characteristics: it is more in the spirit of adaptive in time gradient search techniques; it is somewhat reminiscent of early neural network systems; and it is sometimes useful for rapid adaptation. Speaker adaptation [46], speaker dependent recognition [47], and channel adaptation [48] are all natural extensions of the basic HMM supervised training framework.

Convergence is generally quick in the supervised training scenario of Fig. 6. Three to nine iterations are usually sufficient to capture most of the information in the training database. There are certainly points of diminishing return in training: after a few iterations, recognition performance is usually within 95% of its ultimate performance after many iterations.

Discrimination Techniques

Recently, techniques to improve recognition performance based on notions of discrimination have been introduced in speech recognition. The maximum likelihood approach in training has been shown to be a special case of a design procedure that minimizes the discrimination information [49][3] between the signal probability densities and the HMM probability densities:

$$D(P_S, P_g) = \int p_S(y) \ln(p_S(y)/p_g(y))\, dy, \qquad (30)$$

where P_S denotes the signal probability densities, P_g denotes the HMM probability densities, and where p_S and p_g denote densities for P_S and P_g respectively.

Recognition performance can be improved with the inclusion of a third step in training process that seeks to improve recognition performance by reducing the probability of incorrect recognition hypotheses that compete with the correct choice. One such scheme, originally introduced in isolated word recognition [51], and later extended to continuous speech recognition [52], is known as corrective training. In general, these techniques attempt to identify utterances incorrectly recognized, called the confusion class, and build statistical models that optimally discriminate between the correct class and the confusion class. This has a simple interpretation in the context of HMM training: during training, reduce the probability of all competing incorrect choices, and increase the probability of the correct choice.

Let $c_{ij}(k)$ denote the number of times a transition from state i to state j occurs and symbol v_k is observed. This count can be computed in such a way as to consider the number of competing incorrect word choices, and the number of "near-misses" that occur during recognition of an utterance:

$$c_{ij}(k) = c_{ij}(k) + \gamma(c_{ij}^{+}(k) - c_{ij}^{-}(k)). \qquad (31)$$

$c_{ij}^{+}(k)$ denotes the count corresponding to instances in which utterance recognition was correct. $c_{ij}^{-}(k)$ denotes the counts corresponding to instances in which utterance recognition was incorrect or the probability of incorrect recognition was unacceptably close to the probability of correct recognition, and γ is an adjustment factor that varies between 0 and 1.

An attractive feature of this approach is that discrimination is built directly into the recognition models, and can be incorporated with no additional computational burden during recognition. In continuous speech recognition, identification of the competing choices is somewhat difficult, because incorrect choices often are pruned during the Viterbi beam search. Competing hypotheses can be tracked using stack decoding [51], by examining error patterns for the training database [52], by building grammars that explicitly generate the confusion class [41], or by performing Viterbi beam search for the next best hypothesis [21].

"There is no data like more data." [53]

A central issue in speech recognition is "How much training data is enough?" Most HMM systems today have at least an order of magnitude more free variables than prior DP systems, and require massive amounts of training data. In speaker independent digit recognition, for instance, when recognition performance on an 8,000 utterance database is analyzed at a microscopic level, errors can be associated with particular speaker characteristics not well represented in the training database [41]. Robust parameter estimates from insufficient training data in HMM is a topic unto itself.

The availability of large comprehensive databases has been a significant driving force in speech recognition research. Three important databases (in English) publicly available today, and used extensively in the literature to benchmark HMM performance, are the TIMIT Acoustic Phonetic database [54], the DARPA Resource Management database [55], and the TI/NBS Connected Digit database [56]. The TIMIT database deserves special mention in that it is a speaker independent database designed to cover spoken English, and has been phonetically transcribed and segmented. In addition to these existing database, DARPA recently initiated programs to collect several databases designed to support speech understanding research [57].

EXAMPLES OF HMM SYSTEMS

Let us conclude this paper with a brief overview of several HMM systems[4] that reduced to practice the theory previously described. These systems were selected primarily because they are generally considered to have demonstrated significant advances in the state of the art in speech recognition.

First, we discuss two large vocabulary speech recognition systems based on phonetic models. Much of the original interest in HMM was motivated by the dream that HMM would provide a computationally inexpensive large vocabulary speech recognition framework in which states of an HMM represented some fundamental acoustic unit, such as a phone. HMM technology has at least made phonetic modeling a reality, though the systems described here can certainly not be considered computationally inexpensive. We conclude with a discussion

[3] Minimization of discrimination information is a common theme in statistical signal processing, and is a foundation upon which several approaches to neural nets are based [50].

[4] Naturally, any such review will not do justice to the multitude of speech recognition systems that have served to advance the state of the art.

of speaker independent digit recognition, a small vocabulary problem that challenges the statistical modeling capabilities of HMMs.

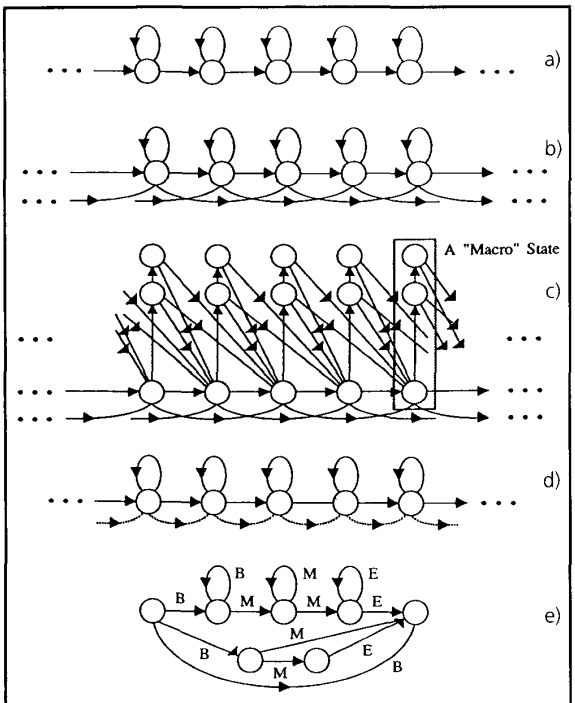


Fig. 9. Some examples of acoustic model topologies that are used today in speech recognition. These models are often referred to as progressive models (or left to right models), since the acoustic match must proceed forward in time. The concept of a progressive model is a straightforward extension of the constrained time alignment procedure in DP.

a) A simple progressive HMM topology. In general, the duration probability density function at a state has an exponential behavior.

b) The Bakis topology (a progressive model with skip states).

c) A finite duration topology. This topology is most analogous to DP.

d) A fenonic baseform topology. The dashed line indicates a transition that produces no output.

e) A modified fenonic baseform with tied transitions. The labels B, M, and E indicate the group to which a transition belongs. Transitions in the same group share output probabilities.

The Tangora System [58]

Perhaps the most visible demonstration of the potential of HMM based speech recognition occurred in the mid 1980s at IBM. The Tangora system [58], a speaker dependent isolated utterance speech recognition system scalable from 5,000 words to 20,000 words was the product of IBM's long-term commitment to applying stochastic modeling to speech recognition. The 5,000 word real-time version of Tangora was implemented using a special purpose processor board occupying a single slot on an IBM PC.

Tangora is based on a VQ front-end (a codebook size of 200) and discrete HMMs. The recognition problem is segregated into an acoustic word matcher and a linguistic decoder. The acoustic word matcher is constrained to matching isolated word utterances. Initially, a Bakis [59] model, shown in Fig. 9(b) was used to describe acoustic models (words) in terms of acoustic vectors (observations). Later, a more flexible representation, denoted denonic baseforms [60], was introduced (see Fig. 9(d)) to reduce the amount of training data required to enroll a new speaker. Each word is represented as a string of phonetic symbols, which in turn are represented as acoustic vectors.

The most unique aspect of the Tangora system is its use of an ngram statistical language model: sentences are described in terms of one, two, and three word combinations (unigrams, bigrams, and trigrams, respectively). The language model has been statistically trained based on a large text database of office correspondence. A spell mode is also incorporated to accommodate user input of words not contained in the predefined language model.

Performance has been measured across a variety of large vocabulary recognition tasks, and is summarized in Fig. 10. Recently, corrective training has been incorporated into the system, reducing the word error rate of a 5,000 word recognition task by 16% [51].

Recognition Task	Word Error Rate (%)
5,000 Word Office Correspondence	2.9%
20,000 Word Office Correspondence	5.4%
2,000 Most Frequent Words In Office Correspondence: Phonetic Baseforms	2.5%
2,000 Most Frequent Words In Office Correspondence: Fenonic Baseforms	0.7%

Fig. 10. Performance of the Tangora system on several speaker dependent large vocabulary recognition tasks [62].

The SPHINX System [35]

Advances in HMM theory have created a renewed focus in speech recognition research on phonetic approaches to large vocabulary speaker independent continuous speech recognition. A good example of the state of the art is the SPHINX system developed at CMU. SPHINX is a speaker independent continuous speech recognizer based on triphone acoustic models (sequences of three phones). Here, we discuss application of SPHINX to the DARPA Resource Management task [55] (a language that consists of a 1000 word vocabulary and a finite state automaton of over 7,000 nodes and 65,000 arcs). A real-time version of SPHINX has been implemented on a VME bus board containing three Weitek general purpose processors.

SPHINX is a VQ-based discrete HMM. A multiple codebook approach for the VQ front-end is used in which cepstral, differential cepstral, and energy features are quantized with separate codebooks [61]. The acoustic models consist of a set of generalized triphone models designed to be as extensible to other applications as possible, yet model as explicitly as possible known coarticulatory behavior. The model topology is shown in Fig. 9(e). A set of 1000 triphones (out of over 7000 that occurred in the training database) were found to be sufficient for the Resource Management task.

Acoustic Model	No Grammar	Word Pair
Monophones with duration modeling	49.6%	83.8%
Monophones with function word and function phrase modeling	59.2%	88.4%
Biphones with function word modeling	67.2%	91.5%
Generalized Triphones	72.8%	94.2%
Generalized Triphones With Between Word Coarticulation	77.9%	95.5%
With Corrective Training	81.9%	96.2%

Fig. 11. Word accuracy of the SPHINX system as a function of the recognition unit [52] on the DARPA Task Domain database. A monophone represents an acoustic unit similar to a phoneme in English. Biphones represent an acoustic unit consisting of a concatenation of two phones. Similarly, generalized triphones represent a unit that is a concatenation of three phones. In the biphone and triphone cases, only those combinations frequently occurring in the DARPA Task Domain database were considered [52]. Performance typically improves as more contextual information is added to the acoustic models. Hence, best performance is obtained with generalized triphones, the largest recognition unit used in these experiments.

Performance for a variety of recognition units is summarized in Fig. 11. A null grammar is a grammar that allows any of the 997 words to follow any other of the word. The word-pair grammar is a grammar that models all permissible sequences of two word combinations. Observe that accuracy improves significantly as the size of language is constrained, and the number of competing choices restricted. Also, Fig. 11 demonstrates that increasing the level of detail in acoustic models results in significant improvements in accuracy (provided there are sufficient training data for the acoustic units).

High Performance Connected Digit Recognition [63]

Speaker independent digit recognition has traditionally been considered a stringent test of a continuous speech recognizer. Grammar constraints are minimal: any digit can follow any other digit. Simultaneously with the evolving HMM large vocabulary systems, research at AT&T was focused on the development of high performance digit recognition using CDHMM. One derivative of this work was a demonstration of a real-time HMM digit recognition system using multiple Digital Signal Processors (DSPs) on the ASPEN multiprocessing system.

Digit recognition systems have traditionally used detailed statistical models [34]. The AT&T digit recognition system [42] is a word-based system that uses progressive models of the form shown in Fig. 9(a). In this system, the statistical model used at each state is a mixture of Gaussian distributions. High performance digit recognition was demonstrated using four models per digit, ten states per model, nine Gaussian distributions (or mixtures) per state, and a unique duration distribution per state. All acoustic models have the same number of states.

In Fig. 12(a), performance, measured as string error rate, is shown as a function of the number of models per digit used. In Fig. 12(b), string error rate is given as a function of the number of mixtures per state. Fig. 12 is based on a digit recognition task consisting of studio quality data [56]. In these experiments, since the elements of the feature set were observed to be essentially uncorrelated with one another, the covariance matrix at each state was assumed to be diagonal.

a)

Number Of Mixtures	Performance
1	9.2%
3	5.8%
5	5.0%
7	4.6%
9	4.2%

b)

Number Of Models	Performance
1	4.35%
2	3.64%
3	3.10%
4	2.94%
5	3.01%
6	3.01%

Fig. 12. Sentence accuracy (sentences are on the average 3.5 digits long) for the AT&T speaker independent digit recognition system on a studio-quality digit recognition task.

a) Sentence accuracy as a function of the number of mixtures per state for the case of 10 states per model with one model per digit.

b) Sentence accuracy as a function of the number of models per digit for the case of 10 states per model with 9 mixtures per state.

The string error rate improves significantly as the number of distributions is increased from one to three, and slightly improves beyond that point. It is interesting to note the similarity between a three mixture state distribution and a three codebook VQ. Error rate also reaches an asymptote at four models per digit. The final recognition performance of this system represented a significant improvement compared to previous DP and acoustic-phonetic approaches.

Improved Digit Recognition [41]

Recently, improved performance on the same digit recognition task described above was demonstrated by Texas Instruments. A discrimination transformation designed to maximize discrimination between the correctly recognized data and the confusion class for each state in each HMM word model. This transformation was applied as an additional transformation in

the Mahalanobis distance of Eqs. 24 and 25. This technique, named phonetic discriminants, is an extension of previous discrimination techniques applied in DP [40], and can be viewed as an alternate approach to corrective training.

Phonetic discriminants were employed in CDHMM system using a finite duration topology similar to that shown in Fig. 9(c). The HMM system used a single multivariate Gaussian distribution per frame in the reference model (the additional states used to create a finite duration structure at each state shared the same observation model).

Several different acoustic modeling techniques were compared ranging from a "pooled-covariance" approach that used a single covariance transformation for all states, to a confusion-discriminant approach that used a unique covariance matrix per state and an additional transformation designed to maximize discrimination. Performance is summarized in Fig. 13. We see a general trend that recognition accuracy improves as the detail of the acoustic model increases. The improvements in Fig. 13 compare favorably with those achieved in corrective training. An implementation of a pooled-covariance system has been demonstrated that uses a single DSP processor.

Acoustic Model	Performance
Pooled Covariance	3.5%
Diagonal Covariance	3.1%
Full Covariance	2.1%
Confusion Discriminants	1.5%

Fig. 13. Sentence accuracy for the TI speaker independent digit recognition system as a function of the acoustic model.

SUMMARY

In this paper, we have reviewed the theory of Hidden Markov models in the context of a continuous speech recognition task. A unified view has been offered in which both linguistic decoding and acoustic matching are treated in an HMM framework. A supervised training paradigm was reviewed that exploits the constraints of the language model to refine recognition models. Several examples of HMM continuous speech recognition systems that represent significant advances in the state of the art were presented.

There are two unmistakable trends surfacing in speech recognition research. First, statistical modeling in speech processing has been elevated to new levels with the introduction of HMM. Our temporal/statistical models of speech are significantly more detailed, resulting in a better acoustic match to the signal for the correct hypothesis. Second, language processing and acoustic processing are more tightly integrated in HMM. It is unclear at this point how much acoustic context is required for accurate acoustic matching, but a trend towards top-down parsing will certainly continue for small language models.

What lies in the future? Again, there are two trends developing. At the acoustic matching level, more general statistical techniques, such as neural nets, are being actively pursued. Key issues here include finding ways to introduce sequential behavior (notions of time) into the system, and finding computationally efficient and data efficient methods of training. At the linguistic level, integration of stochastic representations and higher level grammar formalisms, such as context sensitive grammars, is becoming increasingly important. There is no doubt that synergies of these two will give rise to new, and more powerful, statistical signal processing systems. Rule-governed systems, however, appear to be impractical for unconstrained speech understanding.

Yet at the same time, some basic signal processing problems still remain. Sensitivity to background acoustic noise, changes in the channel and transducer characteristics, and misrecognitions of out of vocabulary responses all remain difficult problems in real speech recognition applications. As our ability to represent real acoustic variation in the signal grows, the need to reject spurious input becomes more acute. While the techniques described here work well in noise-free environments, demonstration of high performance on even the simplest tasks in operational environments remains a challenge.

REFERENCES

[1] S.E. Levinson, "Structural Methods In Automatic Speech Recognition", *Proceedings of the IEEE*, Vol. 73, No. 11, pp. 1625-1650, November 1985.

[2] S.E. Levinson, M.Y. Liberman, A. Ljolje, and L.G. Miller, "Speaker Independent Phonetic Transcription Of Fluent Speech For Large Vocabulary Speech Recognition", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 441-444, Glasgow, Scotland, May 1989.

[3] M. Weintraub, H. Murveit, M. Cohen, P. Price, J. Berstein, G. Baldwin, and D. Bell, "Linguistic Constraints In Hidden Markov Model Based Speech Recognition", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 699-702, Glasgow, Scotland, May 1989.

[4] K. Kita, T. Kawabata, and H. Saito, "HMM Continuous Speech Recognition Using Predictive LR Parsing", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, p. 703-706, Glasgow, Scotland, May 1989.

[5] A. Paeseler and H. Ney, "Continuous Speech Recognition Using A Stochastic Language Model", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 699-702, Glasgow, Scotland, May 1989.

[6] Y.L. Chow and S. Roucos, "Speech Understanding Using A Unification Grammar", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 727-730, Glasgow, Scotland, May 1989.

[7] C. Hemphill and J. Picone, "Speech Recognition in a Unification Grammar Framework", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Proessing*, pp. 723-726, Glasgow, Scotland, May 1989.

[8] A. Weibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme Recognition Using Time-Delay Neural Networks", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 37, No. 3, pp. 328-340, March 1989.

[9] R. Lippman, "An Introduction to Computing With Neural Nets", IEEE ASSP Magazine, Vol. 4, No. 2, pp. 4-22, April 1987.

[10] R.E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.

[11] H. F. Silverman and D. P. Morgan, "The Application of Dynamic Programming To Connected Speech Recognition", *IEEE ASSP Magazine,* elsewhere in this issue.
[12] N. Chomsky, "On Certain Formal Properties of Grammars", *Information Control,* Vol. 2, pp. 137-167, 1959.
[13] M. McMahan and R. B. Price, "Grammar Driven Connected Word Recognition On The TI-SPEECH Board", in *Proceedings Of Speech Tech'86,* pp. 88-91, New York, New York, Apirl 1986.
[14] A. V. Aho and J. D. Ullman, *The Theory Of Parsing, Translation, And Compiling, Volume I: Parsing,* Printice-Hall, Inc., Englewood Cliffs, N.J., 1972.
[15] J. S. Bridle and M. D. Brown, "Connected Word Recognition Using Whole Word Templates", *Proceedings of the Institute of Acoustics, U.K.,* pp. 25-28, 1979.
[16] C. S. Meyers and L. R. Rabiner, "A Level-Building Dynamic Time Warping Algorithm For Connected Word Recognition", *IEEE Transactions On Acoustics, Speech, and Signal Processing,* Vol. 29, No. 2, pp. 284-296, April 1981.
[17] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A Maximum Likelihood Approach To Continuous Speech Recognition", *IEEE Transactions On Pattern Analysis and Machine Intelligence,* Vol. 5, No. 2, pp. 179-190, March 1983.
[18] G. A. Miller, G. A. Heise, and W. Lichten, "The Intelligibility Of Speech As A Function Of The Context Of The Test Materials", Journal of Experimental Psychology, Vol. 41, pp. 329-335, 1951.
[19] B. H. Juang, "On The Hidden Markov Model And Dynamic Time Warping For Speech Recognition-A Unified View", *AT&T Technical Journal,* Vol. 63, No. 7, pp. 1213-1243, September 1984.
[20] S. E. Levinson, "Continuously Variable Duration Hidden Markov Models For Automatic Speech Recognition", *Computers, Speech, and Language,* Vol. 1, No. 1, pp. 29-45, March 1986.
[21] C. H. Lee and L. R. Rabiner, "A Frame Synchronous Network Search Algorithm For Connected Word Recognition", *IEEE Transactions On Acoustics, Speech, and Signal Processing,* Vol. 37, No. 11, pp. 1649-1658, November 1989.
[22] A. Viterbi, "Error Bounds For Convolutional Codes And An Asymptotically Optimum Decoding Algorithm, *"IEEE Transactions On Information Theory,* Vol. 13, No. 2, pp. 260-269, April 1967.
[23] F. Jelinek, "A Fast Sequential Decoding Algorithm Using A Stack", *IBM Journal of Research and Development,* Vol. 13, pp. 675-685, November 1969.
[24] N. J. Nilsson, *Problem-Solving Methods In Artificial Intelligence,* McGraw-Hill, New York, 1971.
[25] P. H. Winston, *Artificial Intelligence,* Addison-Wesley, Reading, Mass., 1984.
[26] L. R. Rabiner and B. H. Juang, "An Introduction To Hidden Markov Models", *IEEE ASSP Magazine,* Vol. 3, No. 1, pp. 4-16, January 1986.
[27] L. R. Rabiner, "A Tutorial On Hidden Markov Models And Selected Applications In Speech Recognition", *Proceedings Of The IEEE,* Vol. 77, No. 2, pp. 257-285, February 1989.
[28] H. Ney, D. Mergel, A. Noll, and A. Paeseler, "A Data-Driven Organization Of The Dynamic Programming Beam Search For Continuous Speech Recognition", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 833-836, Dallas, Texas, April 1987.
[29] B. T. Lowerre and R. Reddy, "The HARPY Speech Understanding System", in W. A. Lea, Editor: *Trends In Speech Recognition,* pp. 340-360, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1980.
[30] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring In The Statistical Analysis Of Probabilistic Functions Of Markov Chains", *Annals of Mathematical Statistics,* Vol. 41, pp. 164-171, 1970.
[31] K. S. Fu, *Syntactic Methods In Pattern Recognition,* Springer-Verlag, New York, 1972.
[32] J. Makhoul, S. Raucos, and H. Gish, "Vector Quantization In Speech Coding", *Proceedings of the IEEE,* Vol. 73, No. 11, pp. 1551-1588, November 1985.
[33] N. Nocerino, F. K. Soong, L. R. Rabiner, and D. H. Klatt, "Comparative Study Of Several Distortion Measures For Speech Recognition", *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 25-28, Tampa, Florida, March 1985.
[34] E. L. Bocchieri and G. R. Doddington, "Frame Specific Statistical Features For Speaker-Independent Speech Recognition", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 34, No. 4, pp. 755-764, August 1986.
[35] K. F. Lee, *Large-Vocabulary Speaker Independent Continuous Speech Recognition: The SPHINX System,* Ph.D. Disertation, Computer Science Department, Carnegie Mellon University, 1988.
[36] B. H. Juang, L. R. Rabiner, and J. G. Wilpon, "On The Use Of Bandpass Liftering In Speech Recognition", *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. 35, No. 7, pp. 947-954, July 1987.
[37] M. R. Anderberg, *Cluster Analysis For Applications,* Academic Press, New York, 1973.
[38] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm For Vector Quantizer Design", *IEEE Transactions On Communications,* Vol. 28, No. 1, pp. 84-95, January 1980.
[39] L. A. Liporace, "Maximum Likelihood Estimation For Multivariate Observations Of Markov Sources", *IEEE Transactions On Information Theory,* Vol. 28, No. 5, pp. 729-734, September 1982.
[40] K. Fukunaga, *Introduction To Statistical Pattern Recognition,* Academic Press, New York, 1972.
[41] G. R. Doddington, "Phonetically Sensitive Discriminants For Improved Speech Recognition", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 556-559, Glasgow, Scotland, May 1989.
[42] L. R. Rabiner, J. G. Wilpon, and F. K. Soong, "High Performance Digit Recognition, Using Hidden Markov Models", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 119-122, New York, April 1988.
[43] L. R. Bahl, P. F. Brown, P. V. DeSouza, and R. L. Mercer, "Maximum Mutual Information Estimation Of Hidden Markov Model Parameters For Speech Recognition", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 49-52, Tokyo, Japan, April 1986.
[44] J. Picone, "On Modeling Duration In Context", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 421-424, Glasgow, Scotland, May 1989.
[45] L. R. Rabiner, C. H. Lee, B. H. Juang, and J. G. Wilpon, "HMM Clustering For Connected Word Recognition", in *Pro-*

ceedings IEEE International Conference on Acoustics, Speech, and Signal Processing, pp 405-408, Glasgow, Scotland, May 1989.
[46] R. Schwartz, Y. Chow, and F. Kubala, "Rapid Speaker Adaptation Using A Probabilistic Spectral Mapping", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 633-636, Dallas, Texas, April 1987.
[47] Y. L. Chow, M. O. Dunham, O. A. Kimball, M. A. Krasner, G. F. Kubala, J. Makhoul, P. J. Price, S. Roucos, and R. M. Schwartz, "BYBLOS: The BBN Continuous Speech Recognition System", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 89-92, Dallas, Texas, April 1987.
[48] A. Nadas, D. Nahamoo, and M. A. Picheny, "Speech Recognition Using Noise-Adaptive Prototypes", *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. 37, No. 10, pp. 1495-1503, October 1989.
[49] Y. Ephraim, A. Dembo, and L. R. Rabiner, "A Minimum Discrimination Information Approach For Hidden Markov Modeling", *IEEE Transactions On Information Theory,* Vol. 35, No. 5, pp. 1001-1013, September 1989.
[50] G. E. Hinton and T. J. Sejnowski, "Learning And Relearning In Boltzmann Machines", in *Parallel Distributed Processing, Volume 1: Foundations,* pp. 282-317, The MIT Press, Cambridge, Massachusetts, 1986.
[51] L. R. Bahl, P. F. Brown, P. V. deSouza, and R. L. Mercer, "A New Algorithm For The Estimation Of Hidden Markov Model Parameters", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 493-496, New York, New York, April 1988.
[52] K. F. Lee, S. Mahajam, "Corrective and Reinforcement Learning For Speaker-Independent Continuous Speech Recognition", Technical Report CMU-CS-89-100, Carnegie Mellon University, January 1989.
[53] R. L. Mercer, "Language Modeling For Speech Recognition", 1988 IEEE Workshop On Speech Recognition, Arden House, Harriman, New York, May 1988.
[54] W. M. Fisher, G. R. Doddington, and K. M. Goudie-Marshall, "The DARPA Speech Recognition Research Database: Specifications and Status", in *Proceedings of DARPA Speech Recognition Workshop,* pp. 93-99, February 1986.
[55] P. J. Price, W. Fisher, J. Bernstein, and D. Pallett, "A Database For Continuous Speech Recognition In A 1000-Word Domain", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 651-654, New York, New York, April 1988.
[56] R. G. Leonard, "A Database For Speaker-Independent Digit Recognition," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 42.11.1-42.11.4, San Diego, California, April, 1984.
[57] J. J. Godfrey, C. T. Hemphill, G. R. Doddington, "The Spoken Language Systems Corpus", to be presented at the DARPA Spoken Language Systems Workshop at Hidden Valley, Pennsylvania, June 1990.
[58] A. Averbuch, L. Bahl, R. Bakis, P. Brown, A. Cole, G. Daggett, S. Das, K. Davies, S. De Gennaro, P. de Souza, E. Epstein, D. Fraleigh, F. Jelinek, S. Katz, B. Lewis, R. Mercer, A. Nadas, D. Nahamoo, M. Picheny, G. Shichman, and P. Spinelli, "An IBM-PC Based Large-Vocabulary Isolated Utterance Speech Recognizer", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 53-56, Tokyo, Japan, April 1986.
[59] R. Bakis, "Continuous Speech Word Recognition Via Centisecond Acoustic States", presented at the 91*st* Meeting of the Acoustical Society of America, Washington, D.C., April 1976.
[60] L. Bahl, P. F. Brown, P. V. deSouza, R. L. Mercer, and M. A. Picheny, "Acoustic Markov Models Used In The Tangora Speech Recognition System", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 497-500, New York, New York, April 1988.
[61] V. N. Gupta, M. Lennig, and P. Mermelstein, "Integration Of Acoustic Information In A Large Vocabulary Word Recognizer", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 697-700, Dallas, Texas, April 1987.
[62] A. Averbuch, L. Bahl, R. Bakis, P. Brown, G. Daggett, S. Das, K. Davies, S. De Gennaro, P. V. deSouza, E. Epstein, D. Fraleigh, F. Jelinek, B. Lewis, R. Mercer, J. Moorhead, A. Nadas, D. Nahamoo, M. Picheny, G. Shichman, P. Spinelli, D. Van Compernolle, and H. Wilkens, "Experiments With The Tangora 20,000 Word Speech Recognizer", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 701-704, Dallas, Texas, April 1987.
[63] L. R. Rabiner, J. G. Wilpon, and F. K. Soong, "High Performance Digit Recognition, Using Hidden Markov Models", *IEEE Transactions On Acoustics, Speech, and Signal Processing,* Vol. 37, No. 8, pp. 1214-1225, August 1989.

Joseph Picone (S'79-M'83-SM'90) received the B.S. in 1979, M.S. in 1980, and Ph.D. in 1983, in electrical engineering, all from the Illinois Institute of Technology, Chicago, Illinois. From 1981 to 1983, he participated in a joint research program at AT&T Bell Laboratories, Naperville, Illinois in the areas of low and medium rate speech coding. In 1983, he joined the Central Research Laboratory at Texas Instruments, where he conducted research into very low rate speech coding and isolated word speech recognition. From 1985 to 1987, he conducted research at AT&T into low rate speech coding for secure communications and isolated word speech recognition for telecommunications applications. In 1987, he returned to Texas Instruments to conduct research into continuous speech recognition. His current interests include large vocabulary speech recognition and integration of speech processing and natural language processing. Dr. Picone has served as an instructor and lecturer at the Illinois Institute of Technology, and is currently an adjunct professor at the University of Texas at Dallas. He has published over 35 papers in the area of speech processing, and has been awarded 2 patents. He is a registered professional engineer in the State of Texas.