A Speaker-Independent, Syntax-Directed, Connected Word Recognition System Based on Hidden Markov Models and Level Building

LAWRENCE R. RABINER, FELLOW, IEEE, AND STEPHEN E. LEVINSON, SENIOR MEMBER, IEEE

Abstract—**In the last several years, a wide variety of techniques have been developed which make practical the implementation and development of large networks for recognizing connected sequences of words. Included among these techniques are efficient and accurate speech modeling methods (e.g., vector quantization, hidden Markov models) and efficient, optimal network search procedures (i.e., level building). In this paper we show how to integrate these techniques to give a speaker-independent, syntax-directed, connected word recognition system which requires only a modest amount of computation, and whose performance is comparable to that of previous recognizers requiring an order of magnitude more computation. In particular, the recognizer we studied was an airlines information and reservation system using a 129 word vocabulary, and a deterministic syntax (grammar) with 144 states, 450 state transitions, and 21 final states, generating more than 6×10^9 sentences. An evaluation of the system, using six talkers each speaking 51 test sentences, yielded a sentence accuracy of about 75 percent resulting from a word accuracy of about 93 percent, for an average speaking rate of about 210 words per minute.**

I. INTRODUCTION

IN the laboratory, techniques for speech recognition have evolved to the point where it is possible to undertake usefully large, complex recognition tasks based on searching networks in a computationally efficient and reliable manner. The earliest such systems included DRAGON and HARPY at Carnegie Mellon University [1], [2], and the IBM System [3]. These recognition systems relied on a single large network to represent all possible sentences in the language, and efficient heuristic search procedures (e.g., the "beam search" for HARPY and the "stack algorithm" for the IBM system) to find, for a given spoken input, the maximum likelihood sentence in the language. An excellent review of the parameters of and performance achieved by these systems is given by Klatt [4].

Somewhat different in spirit, but certainly in the class of "difficult" speech recognition tasks, our own work [5]–[7] has used a smaller vocabulary and a more restrictive syntactic structure in order to allow for speaker independence, telephone inputs, and on-line spoken human–machine conversation. Our system was based on a syntax-directed, level-building search procedure [7], [8] and, although very costly to implement, provided very good performance on tasks related to airline timetable information [7]. A major architectural difference between the airline system and the above cited systems was that its basic recognition unit was the word, whereas units smaller than words were used in HARPY and the IBM system. As a result, the size of the network to be searched in the airline system was considerably smaller than the size of the networks of HARPY or IBM, and thus, an optimal search became feasible.

More recently a great deal of work has been done on learning how to model individual words by hidden Markov models (HMM's) with a small number of states. In particular, by using a discrete symbol representation of the linear predictive coding (LPC) vectors derived from an analysis of the incoming speech, a series of speaker-independent word models has been built for both the digits [9] and the airline [10] vocabularies. Single-word HMM's have previously been proposed by Bakis [11]; however, these models had the number of states comparable to the number of frames in the word and, thus, were different from those used in [9] and [10]. For the airline vocabulary, isolated word error rates on the order of 15 percent were achieved using a word recognition system with an order of magnitude less computation than a conventional dynamic time warping (DTW) approach. We were sufficiently encouraged by these results that we decided to combine the syntax-directed level building search with the isolated word HMM's in an attempt to achieve a robust, accurate, connected word recognizer whose performance compared favorably to that of the DTW approach [10], but with greatly reduced computational load. It is the purpose of this paper to describe our efforts towards this goal.

In the course of implementing the system, we primarily used well-known algorithms for recognition (e.g., Viterbi scoring of HMM's LB on words, etc.). However, we did incorporate two techniques which had not been used previously in our own investigations, namely:

1) a global sentence energy normalization which served to equalize the level of energy peaks during stressed vowels of each word in the sentence;

2) a postprocessor probabilistic word duration model to incorporate some finer temporal constraints into the level building solution.

Thus, this paper is primarily an experimental investigation of the performance of well-known recognition algorithms (with the two small exceptions above) applied to a speaker-independent, telephone line input, task-oriented

Manuscript received June 6, 1984; revised December 1, 1984.
The authors are with AT&T Bell Laboratories, Murray Hill, NJ 07974.

0096-3518/85/0600-0561$01.00 © 1985 IEEE

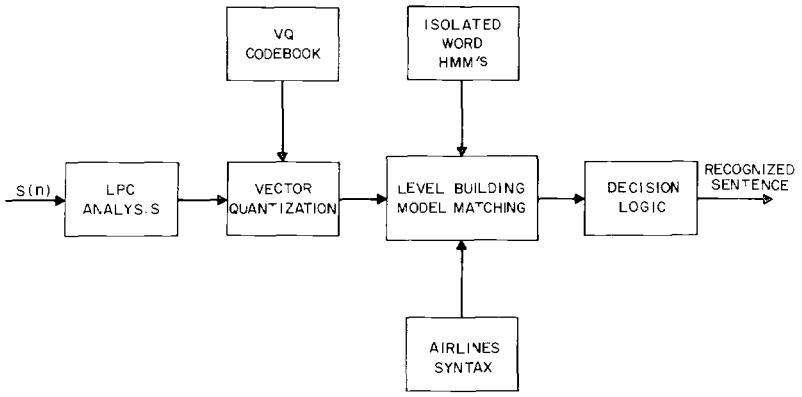

Fig. 1. Block diagram of syntax-directed HMM/LB connected word recognizer.

application. Several experiments were run to investigate the effects on overall system performance of each of the above techniques. Also, a comparison, in terms of both computational complexity and performance, was made between the HMM system and a comparable DTW approach.

The organization of this paper is as follows. In Section II we review the overall structure of the syntax-directed, level-building HMM recognizer. Brief reviews of the various algorithms used to implement the system are given here. In Section III we outline the characteristic of the airline reservation and information system and describe a formal evaluation of the overall recognition system. In this section we present experimental results on both the HMM/LB and DTW/LB systems and discuss the key similarities and differences, and in Section IV we outline areas in which model improvements are required.

II. Overall Structure of the HMM/LB Recognizer

Fig. 1 shows a block diagram of the HMM/LB connected word recognizer. There are essentially four steps in the recognition process involving three externally generated datasets. Thus, the implementation of Fig. 1 requires the specification of six processes, as follows.

1) LPC analysis of the unknown connected word string.

2) Generation of a codebook of M^* vectors for vector quantization of the sequence of LPC vectors of the connected word string.

3) Generation of a set of hidden Markov models for each of the words in the vocabulary.

4) Generation of a syntax (a deterministic grammar) describing how vocabulary words are concatenated to give well-formed sentences in the language.

5) A set of optimal, syntax-constrained matches of sequences of word HMM's to the unknown connected word string. One match for each possible terminal state in the grammar is obtained.

6) Selection of the optimal string.

We now describe, in somewhat more detail, the processing in each stage of the system.

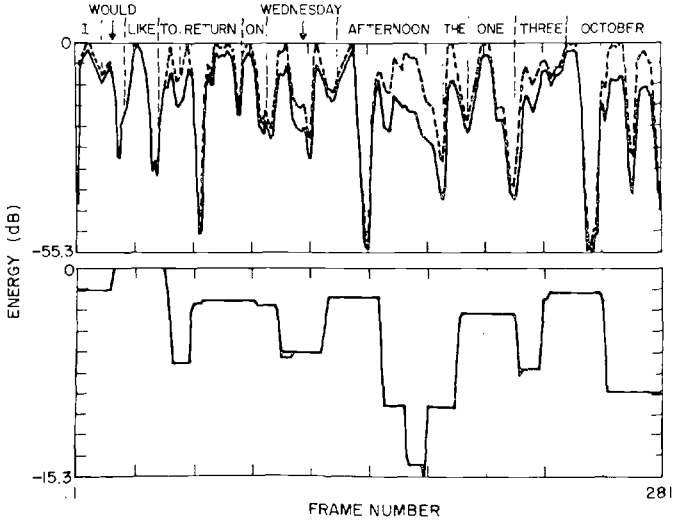

Fig. 2. Original and dynamically normalized energy plots for an airlines reservation sentence. The bottom plot shows the energy correction term as a function of the frame number.

A. LPC Analysis

The LPC analysis is performed as follows. The speech signal $s(n)$, sampled at a 6.67 kHz rate, is first preemphasized by a first-order digital network ($P(z) = 1 - 0.95z^{-1}$). The resulting signal is then blocked into frames of size $N = 300$ samples (45 ms), with adjacent frames separated by $L = 100$ samples (15 ms). Each frame is windowed by a Hamming window and an eighth-order autocorrelation analysis is performed. An eighth-order LPC analysis is then performed and the autocorrelation vector is normalized by the LPC residual. The log energy of each frame is appended as the tenth parameter of the vector for that frame.

A global energy normalization procedure is applied to the entire connected word string [12]. The purpose of the energy normalization is to adjust the sentence energy contour so the peak energy is at or close to 0 dB for each word. In this manner the individual word energy contours can be used to aid in the recognition of words as they occur in the sentence context. Fig. 2 illustrates the effect of the global energy normalization on the log energy contour

for the sentence /I would like to return on Wednesday afternoon the one three October/. The solid curve at the top is the original (unnormalized) log energy contour, and the dashed curve is the normalized log energy contour. The curve at the bottom of the figure is the energy correction applied to the normalized log energy contour to give the unnormalized log energy contour. It can be seen that as much as a 15 dB boost is required to equalize the word peaks to a common 0 dB level. Although the energy normalization provides a significant improvement, it does not always guarantee that the peak energy within a word will reach 0 dB. This is especially true for short function words within a sentence. However, experimentation has shown substantial performance improvements using this energy normalization scheme for some connected word recognition applications [12].

B. Generation of the VQ Codebook

The VQ codebook contains a set of M^* vectors which provide the minimum average distortion (distance) between a given training set of analysis vectors and the codebook entries. In particular we used a codebook with $M^* = 128$ vectors, where each vector was a ten-component vector containing nine residual normalized autocorrelations and a normalized log energy value. The training set consisted of about 100 000 vectors derived from isolated tokens of the words in the vocabulary used for the recognizer. A simple peak log energy normalization, within each word, was used to give the normalized log energy value for each frame. We denote normalized log energy by $\hat{E}$.

If we denote the training set of vectors as $\boldsymbol{a}_i$, $i = 1, 2, \cdots, Q$, and the VQ codebook vectors as $\boldsymbol{b}_j$, $j = 1, 2, \cdots, M^*$, with

$$\boldsymbol{a} = (\boldsymbol{a}_{\mathrm{LPC}}^T, \hat{E}^T) \tag{1a}$$

$$\boldsymbol{b} = (\boldsymbol{b}_{\mathrm{LPC}}^R, \hat{E}^R). \tag{1b}$$

Then the distance between an arbitrary $\boldsymbol{a}$ and an arbitrary $\boldsymbol{b}$ is given by

$$d(\boldsymbol{a}, \boldsymbol{b}) = d_{\mathrm{LPC}}(\boldsymbol{a}, \boldsymbol{b}) + \alpha d_E(\boldsymbol{a}, \boldsymbol{b}) \tag{2a}$$

$$= \left[\frac{(\boldsymbol{b}_{\mathrm{LPC}}^R)^t\, V_T(\boldsymbol{b}_{\mathrm{LPC}}^R)}{(\boldsymbol{a}_{\mathrm{LPC}}^T)^t\, V_T(\boldsymbol{a}_{\mathrm{LPC}}^T)} - 1\right] + \alpha f(|\hat{E}^T - \hat{E}^R|) \tag{2b}$$

where V_T is the matrix of autocorrelations of the LPC system with vector $\boldsymbol{a}_{\mathrm{LPC}}^T$, α is a weighting multiplier on energy distance, and $f(E)$ is a nonlinear energy distance of the form

$$f(E) = \begin{cases} 0 & 0 \le E \le E_{\mathrm{LO}} \\ E - E_{\mathrm{LO}} + E_{\mathrm{OF}} & E_{\mathrm{LO}} > E \le E_{\mathrm{HI}} \\ E_{\mathrm{HI}} - E_{\mathrm{LO}} + E_{\mathrm{OF}} & E_{\mathrm{HI}} < E. \end{cases} \tag{3}$$

Fig. 3 shows a plot of $f(E)$ versus E. Basically, the energy distance is 0 for differences in log energy less than a low threshold (E_{LO}), gives a linear weight (with some possible offset) for log energy differences between E_{LO} and E_{HI}, and clips at some maximum distance for log energy differences greater than E_{HI}. For our applications we used $\alpha = 0.1$, $E_{\mathrm{LO}} = 6$ (dB), $E_{\mathrm{HI}} = 26$ (dB), $E_{\mathrm{OF}} = 6$ (dB) [13].

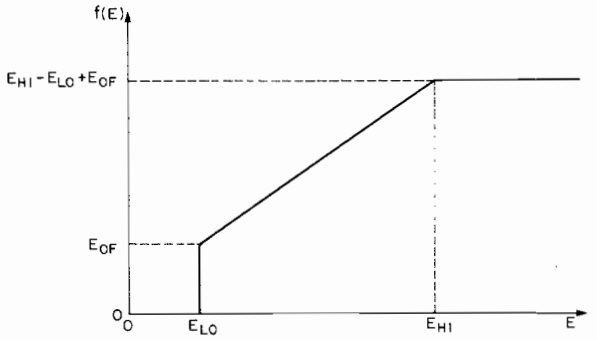

Fig. 3. The nonlinearity used to compute log energy distances.

Given the combined LPC plus energy distance of (2) and (3), the design of the VQ codebook becomes one of choosing the set $\{\boldsymbol{b}_j\}$ such that we satisfy the optimization criterion

$$\bar{D}_{M^*} = \min_{\{\boldsymbol{b}\}} \left[\frac{1}{Q} \sum_{i=1}^{Q} \min_{1 \le j \le M^*} [d(\boldsymbol{a}_i, \boldsymbol{b}_j)]\right]. \tag{4}$$

Various iterative algorithms for approximating the minimization of (4) have been proposed and shown to work quite well over a wide range of conditions [14], [15].

C. Vector Quantization of the Input Analysis Vectors

Once the VQ codebook has been designed, quantization of the input analysis vectors involves computing the distance [according to (3)] between the input vector and each of the M^* codebook vectors, and assigning the index of the codebook vector which gave the minimum distance to the test frame. Thus, for the input vector corresponding to frame t, $\boldsymbol{a}_t$, we compute

$$\hat{d}_j^t = d(\boldsymbol{a}_t, \boldsymbol{b}_j), \qquad j = 1, 2, \cdots, M^* \tag{5a}$$

$$O_t = \underset{1 \le j \le M^*}{\operatorname{argmin}} [\hat{d}_j^t]. \tag{5b}$$

The computation of (5) is performed for all input frames giving the observation sequence O_t, $t = 1, 2, \cdots, T$, where T is the number of frames of speech, and O_t is the index of the VQ codebook entry that best matches the LPC vector for that frame.

D. Generation of Set of Word HMM's

The idea of representing speech events by Markov models has been used in several speech processing systems [1], [11], [16], [17]. Most recently, several researchers have used hidden Markov models to characterize individual words for recognition [9], [10], [18]. The form of the Markov model used to represent each word is shown in Fig. 4. We assume that each word model has N states (where N is typically 5–10) and is characterized by a state transition matrix $A_{(N \times N)}$ and a symbol probability matrix

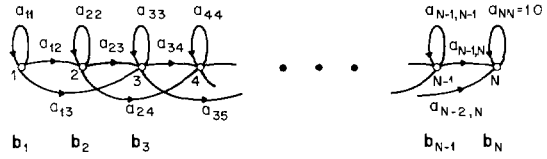

Fig. 4. Hidden Markov model structure for airlines vocabulary words.

$B_{(M^* \times N)}$. For isolated word models we assume A is tridiagonal, i.e.,

$$a_{ij} \neq 0 \quad \text{for } j = i, i+1, i+2$$
$$= 0 \quad \text{otherwise.} \tag{6}$$

The a_{ij}'s form a stochastic matrix, i.e.,

$$a_{ij} \geq 0 \tag{7a}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \quad \forall\, i. \tag{7b}$$

The j, kth entry in B, $b_j(k)$, is the probability of observing symbol k given state j; the $b_j(k)$'s satisfy the stochastic constraint

$$b_j(k) \geq 0 \tag{8a}$$

$$\sum_{k=1}^{M^*} b_j(k) = 1 \quad \forall\, j. \tag{8b}$$

The model parameters a_{ij}, $b_j(k)$, are estimated from a training sequence of R sets of observations (a set of observations corresponds to one utterance of a word) and are used to calculate the probability of the observation set given a particular model M. Reestimation formulas, due to Baum *et al.* [19], were used to iteratively adjust the a_{ij}'s and $b_j(k)$'s until the probability of the observation sequence conditioned on the parameter values stops increasing significantly, or when some other stopping criterion is met (e.g., the number of iterations exceeds some limit).

For the results to be presented in Section III, we used a value of $N = 10$ states for each word model. We also artificially introduced the constraint that for all k, $b_j(k) \geq \epsilon$, i.e., no matter how low the estimated probability of an observation in a state, we clamped it to the minimum value ϵ. Since the $b_j(k)$'s must satisfy the constraint of (8), they had to be renormalized once the ϵ lower bounds were imposed.

E. Formal Syntax

The formal syntax of the language is described by a state transition diagram which defines all word sequences which are considered valid sentences. Fig. 5 shows the state transition diagram for the airline information and reservation system. Myers and Levinson [6] have shown how the states of Fig. 5 (or any other finite state transition diagram without loops) can be topologically sorted so that there can be a transition from state j to state l, only if $j < l$. The use of this procedure is not conceptually important but allows a convenient representation of the grammar in a table organized by initial state, final state, and vocabulary words joining them. This table requires 198 entries to represent the grammar, which has 144 states, 450 transitions, and 21 terminal states. The language specified by the grammar has $>6 \times 10^9$ sentences, each of which is both syntactically and semantically well formed. The sentences range in length from 4 to 22 words, the average (assuming all sentences to be equally likely) is 17 words/sentence. The maximum entropy of the language (allowing for all possible probability distributions of sentences) is 2.15 bits/word [20].

F. Maximum Likelihood String Decoding Using the LB Algorithm

Given the individual word models W_j for each word in the vocabulary, and given a test sequence of observations O_t, $t = 1, 2, \cdots, T$, corresponding to a sequence of words in the vocabulary, the job of the recognizer is to decode O into the sequence of words $\{W_{[1]}, W_{[2]}, \cdots, W_{[P]}\}$ that "best" matches the observation sequence in the sense that the joint probability of the observation sequence and state sequence is maximized.

To understand how we use the LB algorithm to solve for the Viterbi optimal string, consider the case shown in Fig. 6, where we assume each word model is a five-state model, and where we impose no syntactic constraints on the sentences (i.e., any word model can follow any other word model). At level $l = 1$ (the initial level), we begin by matching model $q(W_q)$ to the observation sequence O, beginning at frame 1. To do the match we use a Viterbi decoding of the following form.

1) Initialization—$\delta_1(1) = [b_1^q(O_1)]$, where $\delta_t(j)$ signifies the joint probability of partial state and observation sequences, $\Pr[O_1, O_2, \cdots, O_t$ and $S_1, S_2, \cdots, S_{t-1}, j \mid A, B]$, where S_i is the state at time $t = i$, and

$$\delta_1(j) = 0, \quad j = 2, 3, \cdots, N.$$

2) Recursion, for $2 \leq t \leq T$, $1 \leq j \leq N$

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) * [a_{ij}^q]] * [b_j^q(O_t)]$$

3) Termination—$P(l, t, q) = \delta_t(N)$, $1 \leq t \leq T$

$$B(l, t, q) = 0.$$

According to step 3, at the output of the level we save the result in an array P, which is a function of the level l, the frame number t, and the vocabulary word q.

At level $l = 1$ we cycle through all words in the vocabulary in the manner described above. At the end of the level (when all word models have been used) we level reduce to form the arrays

$$\hat{P}(l, t) = \max_q [P(l, t, q)] \tag{9a}$$

$$\hat{B}(l, t) = B[l, t, \operatorname*{argmax}_q P(l, t, q)] \tag{9b}$$

$$\hat{W}(l, t) = \operatorname*{argmax}_q [P(l, t, q)] \tag{9c}$$

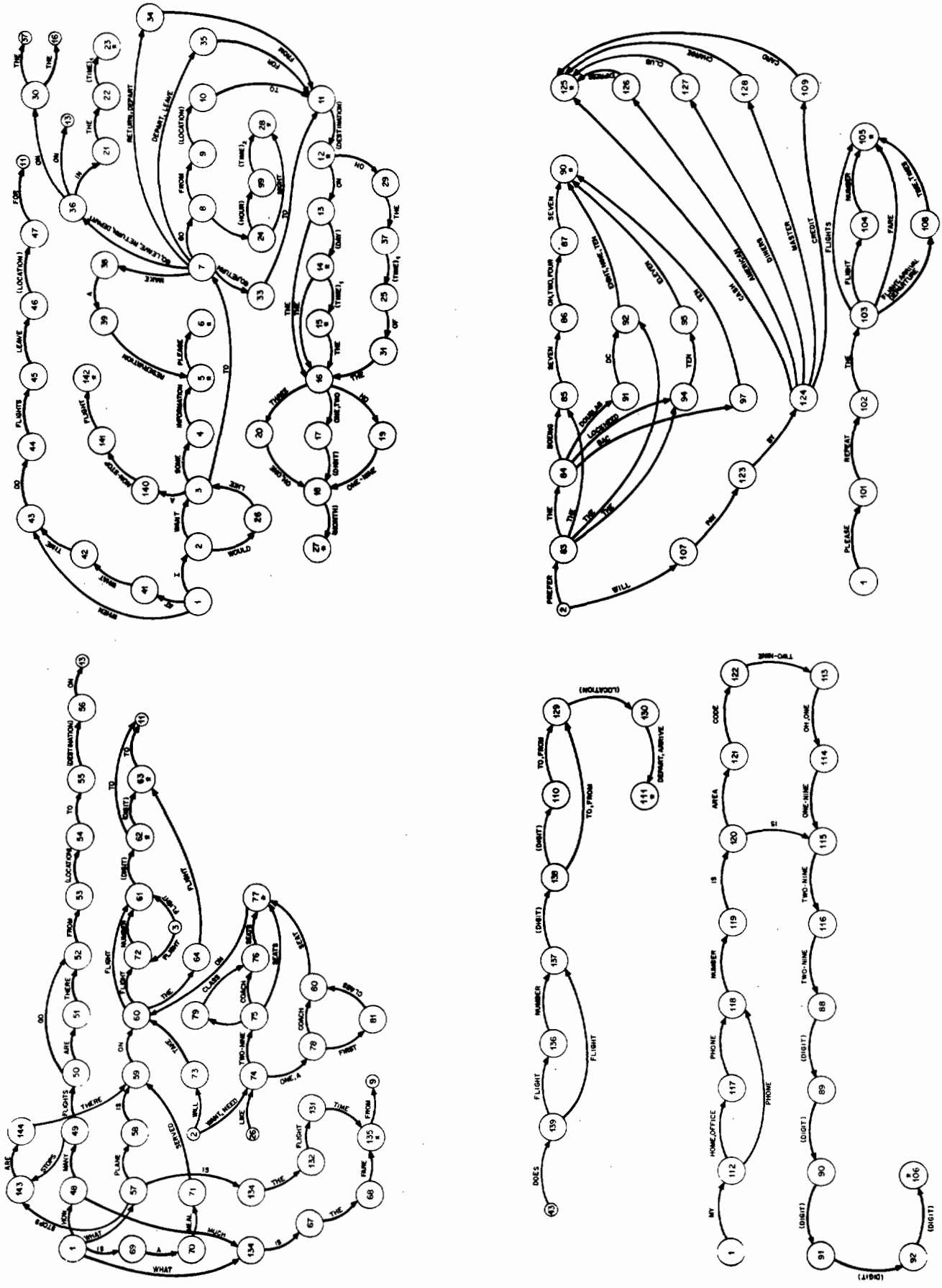

Fig. 5. State transition diagram for the syntax of the airline reservation and information system.

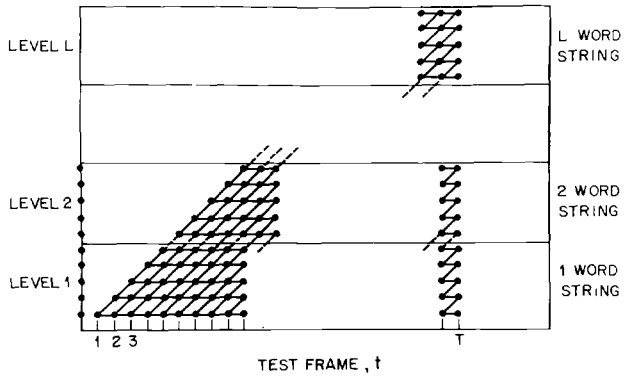

Fig. 6. Implementation of level building based on five-state HMM's for each word.

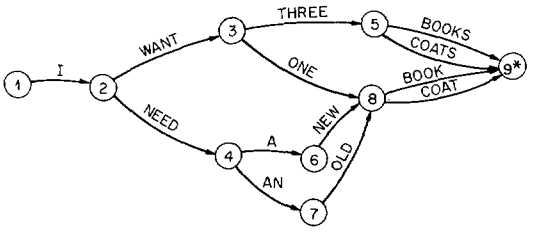

Fig. 7. Illustrative deterministic grammar for showing how grammatical constraints can be incorporated into the LB algorithm.

where $\hat{P}$ is the level output best probability, $\hat{B}$ is the level output backpointer, and $\hat{W}$ is the level output word indicator.

The computation for level 2 (and all higher levels) differs only slightly in the initialization procedure. Since these levels pick up from previous outputs, we have the initialization

$$\delta_1(1) = 0 \tag{10a}$$

$$\delta_t(1) = \max\,[\hat{P}(l-1, t-1), a_{11}^q \delta_{t-1}(1)] \cdot [b_1^q(O_t)], \quad 2 \le t \le T \tag{10b}$$

$$\alpha_t(1) = \begin{cases} t-1 & \text{if } \hat{P}(l-1, t-1) > \delta_{t-1}(1) * a_{11}^q \\ \alpha_{t-1}(1) & \text{otherwise.} \end{cases} \tag{11a}$$

Equation (10a) sets $\delta_1(1)$ to zero; (10b) lets the level pick up at the most appropriate place (or places) from the previous level. Equation (11) creates the appropriate initial backpointer array, which records the frame at the previous level in which the previous word ended. During the recursion (Step 2) the backpointer is updated as

$$\alpha_t(j) = \alpha_{t-1}[\underset{1 \le i \le N}{\operatorname{argmax}} (\delta_{t-1}(i) * a_{ij}^q)] \tag{11b}$$

and at the end of the level, the probability and backpointer arrays become

$$P(l, t, q) = \delta_t(N) \qquad 1 \le t \le T$$

$$B(l, t, q) = \alpha_t(N) \qquad 1 \le t \le T. \tag{12}$$

Once all the word models have been run at any level, the reduced $\hat{P}$, $\hat{B}$, and $\hat{W}$ arrays are formed [using (9)] and the computation proceeds to the next level.

Technically speaking, both $\delta_t(j)$ and $\alpha_t(j)$ are functions of the word index q. However, since the relevant information is saved in the arrays $P(l, t, q)$ and $B(l, t, q)$ [see (12)], the use of an explicit dependence on q in these functions is unnecessary.

The entire procedure terminates when some maximum number of levels L is used. A "best string" of size L words, with probability $\hat{P}(L, T)$, is obtained by backtracking using the backpointer array $\hat{B}(l, t)$ to give the words in the string. The "best" string is then given as the minimum of $\hat{P}(L, T)$ over all possible levels L.

G. Incorporation of Syntax Into LB Algorithm

To modify the LB algorithm described above to handle the syntactic constraints of the sorted state transition diagram is straightforward. One merely makes the association that the levels are uniquely identified with states rather than word position in the string, so that the candidates at the lth level need not be temporally contiguous to those at the $(l+1)$st level. In this manner, the lth level (state), only those models corresponding to words leaving the lth state are matched. In addition, another set of level backpointers is required to link the states (levels) in temporal order. Details of the required modifications are given in [7].

To illustrate the incorporation of syntax into the LB algorithm, Fig. 7 shows a trivial grammar for a 13-word vocabulary consisting of the words

1. I	5. ONE	9. BOOKS	13. OLD
2. WANT	6. A	10. COAT	
3. NEED	7. AN	11. COATS	
4. THREE	8. BOOK	12. NEW	

The terminal state in the grammar is indicated by an asterisk. Typical sentences generated by the grammar include the following.

1) I need a new coat.
2) I need an old book.
3) I want three books.

The correspondence between states in the grammar and levels in the LB algorithm can be seen from the following table of current state, words used, predecessor state, current level, and predecessor levels:

Current State	Words Used	Predecessor State	Current Level	Predecessor Levels
2	I	1	1	0
3	WANT	2	2	1
4	NEED	2	3	1
5	THREE	3	4	2
6	A	4	5	4
7	AN	4	6	4
8	ONE	3	7	2
8	NEW	6	8	5
8	OLD	7	9	6
9*	BOOK, COAT	8	10	7,8,9
9*	BOOKS, COATS	5	11	4

Again, the terminal states (at which a sentence can end) are marked by an asterisk.

This trivial example illustrates the ease with which a deterministic grammar can be incorporated into the LB algorithm.

H. Incorporation of Durational Constraints Into the LB Algorithm

Since the HMM matches to the test string (or some part thereof) are essentially unconstrained temporally, it is possible to expand or contract a model so that it accounts for a large or a small part of the test string, even though words on which the model was trained might never have been as long or as short as the obtained matches. In this manner a word like /Los-Angeles/ could be compressed to match a word like /A/ and vice versa. For a conventional DTW alignment, such problems are eliminated by the global durational constraints built into the algorithm.

To handle three difficulties, a simple Gaussian duration model was assumed for each word in the vocabulary, i.e., the probability density $P_q(D)$ of duration D of the qth word

$$P_q(D) = \frac{\exp - [(D - \bar{D}_q)^2/2\sigma_q^2]}{\sqrt{2\pi}\,\sigma_q} \tag{13}$$

where the mean and standard deviation, $\bar{D}_q$ and σ_q, respectively, were estimated (for each vocabulary word) as the sample mean and standard deviation of the training data used to give the word models.

The way in which the durational model was incorporated into the LB algorithm was as follows. At the end of each level, the word duration d_t, recovered from the backtracking procedure, is given by

$$d_t = t - B(l, t, q) \qquad \forall\, t. \tag{14}$$

Then the cumulative probability at the end of the level was modified to be

$$\tilde{P}(l, t, q) = P(l, t, q) \cdot [P_q(d_t)]^{\gamma} \tag{15}$$

where γ was a weighting parameter that was optimized empirically. The weighted probability scores $\tilde{P}(l, t, q)$ were used in place of the unweighted scores $P(l, t, q)$ in (9), at the end of each level.

Although this method of incorporating durational information is clearly heuristic, in practice it improves the performance significantly by eliminating word candidates that are highly unlikely with respect to temporal information.

I. Choosing the Optimal String

For each terminal state (level) of the grammar (there are 21 such states in the state diagram of Fig. 5), the optimum sentence was traced back. The recognized sentence was the unique word sequence corresponding to the most probable state sequence. For diagnostic purposes we also provided the capability of computing the probability of the actually spoken sentence to learn why an error occurred, and to be able to keep track of statistics on word scores.

TABLE I
WORDS IN THE AIRLINE VOCABULARY

A	A.M.	AFTERNOON
AMERICAN	APRIL	ARE
AREA	ARRIVAL	ARRIVE
AT	AUGUST	B.A.C.
BOEING	BOSTON	BY
CARD	CASH	CHARGE
CHICAGO	CLASS	CLUB
COACH	CODE	CREDIT
D.C.	DECEMBER	DENVER
DEPART	DEPARTURE	DETROIT
DINERS	DO	DOES
DOUGLAS	EIGHT	ELEVEN
EVENING	EXPRESS	FARE
FEBRUARY	FIRST	FIVE
FLIGHT	FLIGHTS	FOR
FOUR	FRIDAY	FROM
GO	HOME	HOW
I	IN	INFORMATION
IS	JANUARY	JULY
JUNE	LEAVE	LIKE
LOCKHEED	LOS-ANGELES	MAKE
MANY	MARCH	MASTER
MAY	MEAL	MIAMI
MONDAY	MORNING	MUCH
MY	NEED	NEW-YORK
NIGHT	NINE	NON-STOP
NOVEMBER	NUMBER	O'CLOCK
OCTOBER	OF	OFFICE
OH	ON	ONE
P.M.	PAY	PHILADELPHIA
PHONE	PLANE	PLEASE
PREFER	REPEAT	RESERVATION
RETURN	SATURDAY	SEAT
SEATS	SEATTLE	SEPTEMBER
SERVED	SEVEN	SIX
SOME	STOPS	SUNDAY
TAKE	TEN	THE
THEE	THERE	THREE
THURSDAY	TIME	TIMES
TO	TUESDAY	TWELVE
TWO	UH	WANT
WASHINGTON	WEDNESDAY	WHAT
WHEN	WILL	WOULD

J. Summary of HMM/LB Recognizer

The sequence of operations used to recognize a spoken sentence are as follows.

1) LPC analysis (including sentence dynamic energy normalization).

2) VQ in which the unknown utterance is represented as a sequence of observations from a finite symbol vocabulary (the set of VQ codebook entries).

3) A syntax-directed LB match to the sentence, one state at a time, based on a finite state transition diagram of the language.

4) Selection of the Viterbi optimal sentence among the set of sentences corresponding to terminal states (levels) in the system.

In the next section we outline a set of experiments that were performed to evaluate the performance of the HMM/LB recognizer.

III. EVALUATION OF THE SYNTAX-DIRECTED HMM/LB RECOGNIZER

The specific task to which we applied the syntax-directed, HMM/LB recognizer was the airlines information and reservation system [5]–[7]. For this system a vocabulary of 129 words (Table I) was required. This vocabulary includes very many highly confusable words, including the sets (DC, BAC), (MAY, MANY), (DO, TO, TWO), (AM, PM), (IS, IN), (TIME, TIMES), (SEAT, SEATS), (FROM, SOME), (CODE, COACH), (WANT, WHAT), (NIGHT, FLIGHT, FLIGHTS), (A, PAY), (MUCH, MARCH), (I, BY, MY), (THEE, THREE), etc. Previous

TABLE II
SENTENCES USED TO EVALUATE THE AIRLINE RECOGNITION SYSTEM

1	I WANT TO MAKE A RESERVATION
2	I WOULD LIKE SOME INFORMATION PLEASE
3	I WANT TO GO FROM NEW-YORK TO LOS-ANGELES ON TUESDAY MORNING
4	I WOULD LIKE TO RETURN ON WEDNESDAY AFTERNOON THE ONE THREE OCTOBER
5	I WOULD LIKE A NON-STOP FLIGHT
6	WHEN DO FLIGHTS LEAVE PHILADELPHIA FOR DETROIT ON MONDAY AFTERNOON
7	I WANT TO GO AT TWELVE O'CLOCK
8	I WOULD LIKE TO DEPART AT NIGHT
9	I WANT TO LEAVE IN THE MORNING
10	I WANT TO DEPART FROM BOSTON ON THE EVENING OF THE OH NINE NOVEMBER
11	HOW MANY FLIGHTS ARE THERE FROM WASHINGTON TO DENVER ON THURSDAY NIGHT
12	HOW MANY FLIGHTS GO FROM SEATTLE TO MIAMI ON THE TWO EIGHT FEBRUARY
13	WHAT PLANE IS ON FLIGHT TWO SIX TO CHICAGO
14	HOW MANY STOPS ARE THERE ON THE FLIGHT
15	I WOULD LIKE FLIGHT NUMBER FOUR ONE
16	I WILL TAKE FLIGHT FIVE THREE
17	I WOULD LIKE A FIRST CLASS SEAT
18	I NEED THREE SEATS
19	I WANT ONE COACH SEAT
20	WHAT IS THE FLIGHT TIME FROM BOSTON TO CHICAGO
21	IS A MEAL SERVED ON THE FLIGHT TO DENVER
22	HOW MUCH IS THE FARE
23	WHAT IS THE FARE FROM DETROIT TO PHILADELPHIA ON SUNDAY NIGHT
24	WHEN DOES FLIGHT NUMBER TWO FROM LOS-ANGELES ARRIVE
25	AT WHAT TIME DOES FLIGHT SEVEN ONE TO SEATTLE DEPART
26	MY HOME PHONE NUMBER IS AREA CODE TWO OH ONE SIX TWO FOUR ONE TWO FOUR SIX
27	MY OFFICE PHONE NUMBER IS FIVE THREE SIX TWO ONE FIVE TWO
28	PLEASE REPEAT THE ARRIVAL TIMES
29	PLEASE REPEAT THE DEPARTURE TIME
30	I WILL PAY BY CREDIT CARD
31	I PREFER THE LOCKHEED TEN ELEVEN
32	I PREFER THE BOEING SEVEN FOUR SEVEN
33	I PREFER THE D.C. NINE
34	I PREFER THE DOUGLAS D.C. TEN
35	I PREFER THE B.A.C. TEN
36	I WILL PAY BY MASTER CHARGE
37	I WILL PAY BY CASH
38	I WILL PAY BY DINERS CLUB
39	I WILL PAY BY AMERICAN EXPRESS
40	I WANT TO GO AT ELEVEN A.M.
41	I WANT TO GO AT SIX P.M.
42	I WANT TO RETURN TO CHICAGO ON THE THREE OH DECEMBER
43	I WOULD LIKE TO DEPART ON FRIDAY EVENING
44	I WOULD LIKE ONE FIRST CLASS SEAT ON FLIGHT NUMBER FOUR FOUR TO LOS-ANGELES
45	I WANT TO RETURN ON THE OH NINE MARCH
46	I WANT TO GO TO WASHINGTON ON THE TWO FOUR APRIL
47	I WOULD LIKE TO RETURN TO NEW-YORK ON THE OH ONE MAY
48	I WANT TO LEAVE FOR LOS-ANGELES ON THE MORNING OF THE ONE FOUR JUNE
49	I WANT TO GO FROM BOSTON TO PHILADELPHIA ON TUESDAY MORNING THE OH FOUR JULY
50	I WOULD LIKE TO RETURN ON THE OH SEVEN AUGUST
51	AT WHAT TIME DO FLIGHTS LEAVE BOSTON FOR DENVER ON THE TWO SEVEN SEPTEMBER

experimentation has shown that an isolated word recognition accuracy of about 88 percent in a speaker-trained mode and 91 percent in a speaker-independent mode is obtainable with the vocabulary [21], [22]. (Interestingly, since the speaker-independent mode used 12 templates per word, it was able to achieve slightly higher performance than the speaker-dependent mode, which used one template per word.)

The grammar, or syntax, for the system is the one given in Fig. 5. In the state transition diagram, there are 144 distinct states with 450 state transitions, 21 final states, and a capability of generating about 6×10^9 distinct sentences. When topologically sorted into the format convenient for implementation via the level building algorithm, we get 198 equivalence classes of state transitions (i.e., those having the same initial and final state).

A. Database Used in the Evaluation

To evaluate the performance of the system of Section II, a set of six talkers (five male, one female—all experienced users of speech recognition systems) each spoke a set of 51 sentences from the airlines grammar (see Table II). The sentences were carefully chosen to include every vocabulary word and every state transition in the language. The talkers were asked to read each sentence in a natural manner; no instructions as to speed or manner of articulation were given to the talkers.

Four of the six talkers also individually trained the system by using a robust training procedure [23] to provide speaker-dependent isolated word reference patterns. Using the robust training procedure, a single, isolated-word reference pattern was obtained for each word of the vocabulary. These patterns were used in a speaker-dependent evaluation of the DTW/LB implementation of the airline system. No attempt was made to train speaker dependent HMM's for these four talkers.

The speaker-independent models (and also the templates) were derived from an earlier training set of 100 talkers (50 male, 50 female), each of whom used the robust training procedure to provide a single robust isolated word token for each of the 129 words in the vocabulary [22]. The six subjects who provided the test set were not among the 100 from whom the training data were taken. For the DTW implementation, a clustering analysis [24] of the 100 tokens of each word provided a set of 12 speaker-independent templates for each word in the vocabulary, where the templates were created as the cluster center obtained by averaging for each of the 12 largest clusters. (Template sets both with and without frame energy were created.)

TABLE III
AVERAGE RATE AND RATE RATIO STATISTICS FOR THE SIX TEST TALKERS

TALKER	AVERAGE RATE (WPM) TEST	S.D. References	S.I. References	RATE RATIO S.D. Test	S.I. Test
LR	184	131	108	0.72	0.59
JW	197	134	108	0.69	0.56
SL	210	158	108	0.77	0.52
AR	234	178	108	0.77	0.47
CS	201	—	108	—	0.54
DK	231	—	108	—	0.47

Similarly, for the HMM's, the 100 training tokens of each word (suitably vector quantized) were used as input to the Baum–Welch reestimation procedure to provide a single HMM for each vocabulary word. After much preliminary experimentation with the isolated word models [10], a value of $N = 10$ states, with $M^* = 128$ VQ codebook symbols, was used for all models.

Table III gives several average statistics of the test sequences and the DTW reference patterns. Included in the table, for each of the six test talkers, are the average talking rates (in words per minute—wpm) of the test sequences, the speaker dependent (SD) reference pattern rate (i.e., strung together in the appropriate sequence for each test sentence), the average speaker independent (SI) reference pattern rate (i.e., stringing together an average length template for each word in the string), the ratio of rates of the speaker-dependent patterns to the test pattern, and finally, the ratio of rates of the speaker-independent patterns to the test pattern. Several interesting observations can be made from the data of Table III. First we see that the average talking rate of the test utterances was indeed very high, and in fact exceeded 200 wpm for four of the six talkers. Next we see a high degree of variability in the average rate of the SD references across the six talkers. The synchrony of the SD references with the tests is seen in that the rate ratios for the SD/TEST are almost constant. The rate ratios of the SI/TEST are also fairly uniform; however, the ratios hover perilously close to 0.5 (in fact, they fall below it for two of the six talkers), indicating potential problems due to violations of DTW temporal constraints. Of course, Table III reflects only average SI templates, whereas there is considerable variability in the rate ratios for the SI case in using individual SI templates.

It is worthwhile contrasting the test set used here with a previous one used by Myers and Levinson [7]. Myers and Levinson (with essentially the same sentences) had an average rate of about 171 wpm as opposed to the 210 wpm average rate here. Since the same SI reference patterns were used in both experiments, we expect (and will see later) that the performance here will be somewhat poorer than that achieved in the earlier study.

B. Recognition Experiments

The test database (6 talkers × 51 sentences) was run on the HMM/LB and the DTW/LB recognizers. For the DTW/LB we varied only one condition, namely, whether or not the sentence energy contour was used. The DTW/LB system was run in both speaker-trained (1 template/word) and speaker-independent (12 templates/word) conditions.

TABLE IV
(a) RESULTS ON DTW/LB RECOGNIZER FOR BOTH SD AND SI RUNS. (b) INDIVIDUAL TALKER RESULTS FOR DTW/LB SD BEST RUN. (c) INDIVIDUAL TALKER RESULTS FOR DTW/LB BEST RUN.

MODE	USE ENERGY	PERCENTAGE OF SENTENCES WITH β OR FEWER WORD ERRORS 0	1	2	3	4	5	6	7	WORD ERROR RATE %
SD	NO	82.4	89.2	94.1	96.1	97.1	98.5	98.5	98.5	3.9
SD	YES	86.9	89.2	94.6	97.6	99.0	100	100	100	3.8
SI	NO	73.1	84.8	91.2	94.6	96.0	98.7	99.0	100	7.1
SI	YES	74.2	79.7	93.5	95.4	96.4	98.0	98.7	99.7	7.5

TALKER	NUMBER OF SENTENCES WITH β WORD ERRORS 0	1	2	3	4	5	6	7
LR	50		1					
JW	35		9	5	1	1		
SL	47	2	1		1			
AR	45	3		1	1			

TALKER	NUMBER OF SENTENCES WITH β WORD ERRORS 0	1	2	3	4	5	6	7
LR	44	2	5					
JW	41	2	7	1				
SL	46	2	2		1			
AR	31	3	9	1	1	2		3
CS	32	6	9	2	1		1	
DK	33	2	10	2		3	1	

For the HMM/LB recognizer, a series of runs was made to determine the syllabic rate smoothing parameters of the dynamic energy normalization procedure, and then a series of runs were made with variable values of γ [see (15)], the duration model weighting parameter, and finally, a pair of runs was made to show the effectiveness of the dynamic energy normalization procedure. We also performed one pair of runs on the isolated word HMM recognizer on an independent isolated word test set to see if the durational model could help the isolated word case.

C. Recognition Results—DTW/LB

The results of running the test set through the syntax-directed, DTW/LB recognizer, in both speaker-dependent and speaker-independent modes, are given in Table IV(a). Each sentence was scored and the number of word errors (from 0 to 7) was recorded for each talker. Table IV(b) and (c) gives breakdowns by individual talker for the speaker-dependent and speaker-independent results, respectively. In tabulating the number of word errors in a sentence, insertions were essentially disregarded in that they did not add to the word error count. However, *all* sentences with insertions did have one or more word errors; hence, the overall sentence error rate is unaffected. A word was judged in error if it did not appear in the recognized sentence in the sequence in which it was spoken.

An analysis of the results given in Table IV shows the following:

1) In the SD mode, the use of energy leads to a 4.5

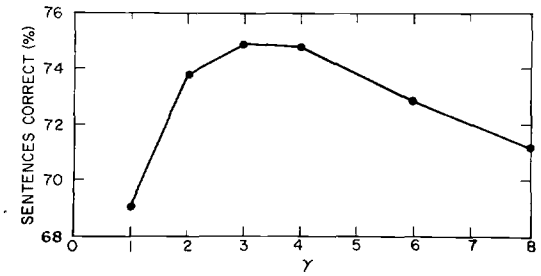

Fig. 8. Plot of percentage sentences correct versus duration weight γ, for the HMM/LB recognizer.

percent increase in overall sentence accuracy. It can be seen that most of this increase comes from sentences that, without energy, had single word errors.

2) In the SI mode, the use of energy leads to a modest 1.1 percent increase in overall sentence accuracy. It can be seen that the percentage of sentences with one or fewer word errors is considerably smaller (5.1 percent) for the case with energy than without. This is an unusual phenomenon due to the high propensity of the system to change /would like/ to /want/ and, hence, incur a double word error in the sentence. Since the sequence /would like/ occurs nine times out of the 51 sentences, and since the energy contours of /would/ followed by /like/ are, in general, grossly different from /would like/ when spoken in context, a large number of the double word errors in the SI case involved this sequence. It should be noted that, in all such cases, the recognized sentence was semantically correct.

3) For the four talkers who provided the SD results, one talker performed significantly better in the SI mode than in the SD mode, one talker was about the same in both modes, and two talkers performed significantly better in the SD mode.

D. Recognition Results—HMM/LB

The first series of runs varied the probability duration model weighting factor γ [see (15)] and measured the overall sentence recognition accuracy. Results of these runs are shown plotted in Fig. 8. It can be seen that the best performance is obtained for a weighting factor of $\gamma = 3$. At this value of γ, the sentence accuracy is 5.9 percent higher than for a value of $\gamma = 1$. Furthermore, the performance is 12.8 percent higher than for the case when no duration model was used at all (i.e., $\gamma = 0$)! Hence, it should be clear that, even at the word level, the duration model provides a powerful recognition factor for this system.

Because of the power of the word duration model in the connected word application, it was also applied directly to the isolated word system for the airlines vocabulary. Fig. 9 shows a comparison plot of the error rate versus the top recognition candidates (i.e., whether the correct word was at top one candidate, within the top two candidates, etc.) for the isolated word HMM recognizer both with and without the duration model. It can be seen that even in the isolated word case, a modest but consistent improvement

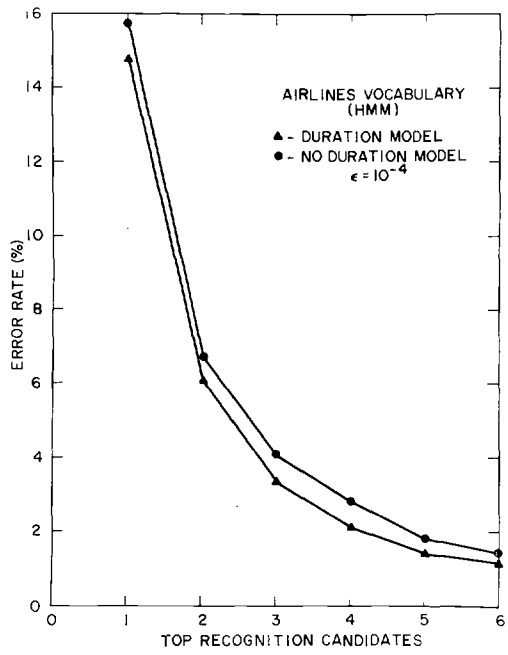

Fig. 9. Plots of error rate (percent) versus number of top recognition candidates for the isolated word airlines vocabulary both with and without the probability duration model.

TABLE V
(a) RESULTS ON HMM/LB RECOGNIZER FOR SI RUNS. (b) INDIVIDUAL TALKER RESULTS FOR HMM/LB BEST RUN.

USE DYNAMIC ENERGY NORMALIZATION	PERCENTAGE OF SENTENCES WITH β OR FEWER WORD ERRORS								WORD ERROR RATE (%)
	0	1	2	3	4	5	6	7	
NO	66.0	81.4	87.9	91.5	94.8	98.7	99.7	100	9.3
YES	74.9	86.3	90.5	94.4	97.1	99.4	100	100	6.7

TALKER	NUMBER OF SENTENCES WITH β WORD ERRORS							
	0	1	2	3	4	5	6	7
LR	42	7	1		1			
JW	44	5		1		1		
SL	47	2		1	1			
AR	28	7	5	4	4	3		
CS	38	8	2	1		1	1	
DK	30	6	5	5	2	2	1	

in word recognition accuracy of about 1 percent is observed for the top six candidate positions.

Using the weight of $\gamma = 3.0$, the next experiment compared the performance of the overall system both with and without dynamic energy normalization of the test sentences. The results of this experiment are given in Table V. (Recall that here we are discriminating only whether or not we dynamically normalize the test energy contour. It has been previously shown that energy is a necessary parameter for the airlines system [10] using HMM's, and without it the sentence accuracy is on the order of 5–10 percent.) It can be seen that the dynamic energy normalization procedure increases the sentence accuracy by 8.9 percent and decreases the word error rate by 2.6 percent. These improvements in performance are statistically significant at any reasonable level of significance.

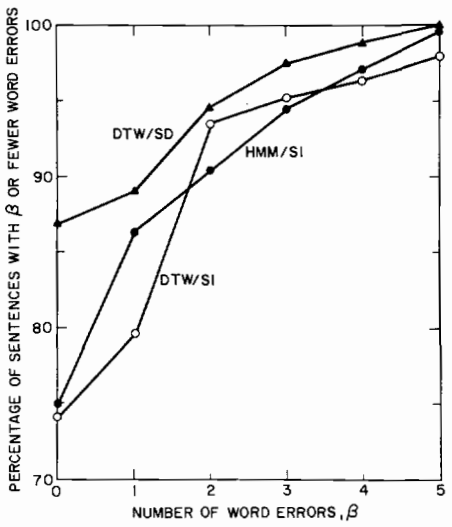

Fig. 10. Plots of percentage of sentences with β or fewer word errors versus the number of word errors β, for the HMM/LB and the DTW/LB systems.

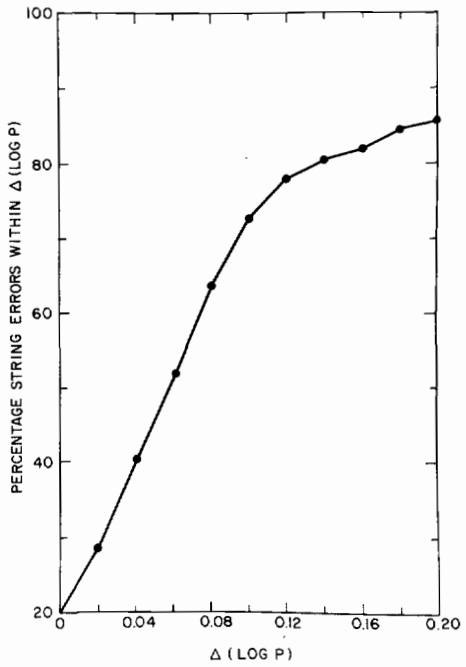

Fig. 11. Plot of percentage string errors within $\Delta(\log P)$ versus $\Delta(\log P)$ for the HMM/LB system.

E. Comparison of DTW/LB and HMM/LB Results

Fig. 10 shows a plot of the percentage of sentences with β or fewer word errors as a function of the number of word errors β in the sentence for the best DTW/LB SD and SI and HMM/LB SI results. The curves show virtually identical performance for both the DTW and HMM implementations in the SI mode, and a significant performance improvement for the SD implementation. An examination of the results in Tables IV and V shows that the four talkers used in the SD runs gave much better performance, on average, than the other two talkers used in the SI runs. Thus, the difference in performance between the SD and SI runs, as shown in Fig. 10, is not really as large as shown; on the basis of using the same four talkers in each case, about half the difference in accuracy is accounted for.

F. Some Additional Analysis Results

To better understand the performance of the recognizers on the airlines reservation task, several additional analyses of the SI experimental results were made. One analysis tried to answer the question of how far away (in a log probability sense) were the actually spoken sentences when the system misrecognized the input—i.e., was an error made because of a poor fit to the correct references, or because of a better fit to incorrect references. In order to gain this understanding, the HMM/LB and DTW/LB recognizers were rerun on all the test data, but using a degenerate syntax such that only the actually spoken sentence was scored. In this manner we could compare the average log probability (or distance) per observation for the incorrect sentence versus that for the actually spoken sentence, and ascertain how close the correct sentence actually was. To quantify these data, Fig. 11 shows a plot of the percentage of the sentence errors (out of the 77 sentence errors made from 306 sentences for the HMM/LB system) whose correct sentence distance was within $\Delta(\log P)$ from the actually recognized sentence, as a function of $\Delta(\log P)$. (A similar curve could be constructed for the DTW/LB system with $\Delta(D)$ scores.) Since the average log probability per observation ranged from -3 to -4.5 for these sentences, it can be seen that, in general, the vast majority (close to 75 percent) of the incorrect sentences had scores within a very small amount (0.10) of the score for the actual sentence. These cases generally represent sentences with a small number of word errors (1–3) which had errors in days of the week, digits in a string, /want/ versus /would like/, etc. The remaining 15 or so sentence errors (i.e., about 25 percent of them) represent globally bad matches and have average $\Delta(\log P)$ scores of greater than 0.10.

A second analysis was made of all words whose average log probability scores exceeded some large threshold. The word log probability scores were obtained by backtracking each sentence, knowing the correct string, and obtaining individual word scores from the backtracking. This analysis was done for both the HMM/LB and DTW/LB SI runs. The results of this analysis showed the following:

1) For the HMM/LB system, the words /TO/, /THE/, and /WOULD/ had a large number of occurrences where the average log probability per frame exceeded -5.0. In particular, these events occurred 26 times for /TO/, 18 times for /THE/, and seven times for /WOULD/. No other word had more than three occurrences of large scores throughout the test.

2) For the DTW/LB system, the words /I/, /THE/, /TO/, and /WOULD/ had a large number of occurrences where the average frame distance exceeded 1.0. In particular, these events occurred 28 times for /I/, 26 times for /THE/, 22 times for /TO/, and seven times for /WOULD/.

No other word had more than four occurrences for large distances throughout the test.

The degree of overlap of these poorly performing words emphasizes a major problem with the recognizer—namely, that the word models (templates) were derived entirely from isolated word tokens. For content words such as /LOS-ANGELES/ or /RESERVATION/, there is very little problem using isolated word tokens for recognition of these words in context; for highly variable function words such as /TO/, /THE/, /WOULD/, etc., the context often changes the words significantly, so that the isolated word models are totally inappropriate. Perhaps the major reason why the entire recognition system performs as well as it does is because there are many content words, and as long as those are properly recognized, the system is quite forgiving of gross misfits on the function words.

IV. Discussion and Summary

The results presented in the previous section show that the HMM/LB implementation of the airlines reservation and information system performs comparably to the DTW/LB implementation of the same system in the SI modes. This result is somewhat remarkable for two reasons, namely:

1) the HMM/LB system undergoes fairly severe distortion in using a VQ to provide the discrete observation sequence—the DTW/LB has no such quantization distortion [10];

2) the HMM/LB system requires an order of magnitude less computation than the DTW/LB system.

The comparable performance of the two systems was shown to be related to the use of a robust, heuristic word duration probability model in the LB implementation, and to the use of a dynamic energy normalization algorithm which served to highlight content words in the string.

The error analysis indicated two problems with the system. First, there are severe problems with the word models for some of the highly variable function words in the vocabulary. Second, there are some fine analysis problems in resolving close-sounding words—e.g., days of the week. The solution to both these problems is straightforward. The only proper way to train a connected word recognizer is from word tokens derived from connected word sequences spoken directly to the recognition system. These connected word sequences can then be used to help train the system to properly recognize embedded word tokens.

A second potential system improvement, for the HMM/LB system, is to eliminate VQ from the system, i.e., to use continuous distributions, rather than discrete symbols in the HMM's. Preliminary investigations indicate that this is indeed a viable possibility and one that must be considered in future research. The durational model is also amenable to refinement by the use of variable duration HMM's [25].

Work at IBM [26] has indeed been using all three proposed system refinements and has achieved excellent results on a wide-band speech, single-talker system with a vocabulary of 250 words. However, it remains to be seen how the degradations of telephone quality speech and the need for speaker independence will affect the resulting system performance on the task studied by the IBM researchers.

In summary, we have shown that a fairly reliable, syntax-directed, connected word recognition system with a medium-size vocabulary can be efficiently implemented using HMM's and the LB algorithm.

References

[1] J. K. Baker, "The DRAGON system—An overview," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 24–29, Feb. 1975.

[2] B. T. Lowerre and D. R. Reddy, "The Harpy speech understanding system," in *Trends in Speech Recognition*, W. Lea, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1980, pp. 340–360.

[3] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-5, pp. 179–190. Mar. 1983.

[4] D. H. Klatt, "The problem of variability in speech recognition and in models of speech perception," in *Variability and Invariance of Speech Processes*, J. Perkell and D. H. Klatt, Eds. Hillsdale, NJ: Erlbaum, to be published.

[5] S. E. Levinson, "The effects of syntactic analysis on word recognition accuracy," *Bell Syst. Tech. J.*, vol. 57, pp. 1627–1644, May–June 1978.

[6] S. E. Levinson and K. L. Shipley, "A conversational-mode airline information and reservation system using speech input and output," *Bell Syst. Tech. J.*, vol. 59, pp. 119–137, Jan. 1980.

[7] C. S. Myers and S. E. Levinson, "Speaker-independent connected word recognition using a syntax-directed dynamic programming procedure," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 561–565, Aug. 1982.

[8] C. S. Myers and L. R. Rabiner, "A level building dynamic time warping algorithm for connected word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 284–296, June 1981.

[9] L. R. Rabiner, S.E. Levinson, and M. M. Sondhi, "On the application of vector quantization and hidden Markov models to speaker-independent, isolated word recognition," *Bell Syst. Tech. J.*, vol. 62, pp. 1075–1105, Apr. 1983.

[10] ——, "On the use of hidden Markov models for speaker independent recognition of isolated words from a medium size vocabulary," *AT&T Bell Lab. Tech. J.*, vol. 63, pp. 627–642, Apr. 1984.

[11] F. Jelinek, "Continuous speech recognition by statistical methods," *Proc. IEEE*, vol. 64, pp. 548–549, Apr. 1976.

[12] L. R. Rabiner, "On the application of energy contours to the recognition of connected word sequences," *AT&T Bell Lab. Tech. J.*, vol. 63, pp. 1981–1995, Nov. 1984.

[13] L. R. Rabiner, M. M. Sondhi, and S. E. Levinson, "A vector quantizer combining energy and LPC parameters and its application to isolated word recognition," *AT&T Bell Lab. Tech. J.*, vol. 63, pp. 721–736, May 1984.

[14] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantization," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.

[15] B. Juang, D. Wong, and A. H. Gray, Jr., "Distortion of vector quantization for LPC voice coding," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 294–303, Apr. 1982.

[16] A. B. Poritz, "Linear predictive hidden Markov models and the speech signal," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Paris, France, May 1982, pp. 1291–1294.

[17] F. Jelinek, L. R. Bahl, and R. L. Mercer, "Design of a linguistic decoder for the recognition of continuous speech," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 250–256, May 1975.

[18] R. Billi, "Vector quantization and Markov source models applied to speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Paris, France, May 1982, pp. 574–577.

[19] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Statist.*, vol. 41, pp. 164–171, 1970.

[20] M. M. Sondhi and S. E. Levinson, "Computing relative redundancy to measure grammatical constraint in speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tulsa, OK, Apr. 1978, pp. 409–412.

[21] S. E. Levinson and A. E. Rosenberg, "A new system for continuous speech recognition—Preliminary results," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Washington, DC, Apr. 1979, pp. 239–243.
[22] J. G. Wilpon, L. R. Rabiner, and A. Bergh, "Speaker-independent isolated word recognition using a 129-word airline vocabulary," *J. Acoust. Soc. Amer.*, vol. 72, pp. 390–396, Aug. 1982.
[23] L. R. Rabiner and J. G. Wilpon, "A simplified robust training procedure for speaker trained, isolated word recognition systems," *J. Acoust. Soc. Amer.*, vol. 68, pp. 1271–1276, Nov. 1980.
[24] L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, "Speaker independent recognition of isolated words using clustering techniques," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 336–349, Aug. 1979.
[25] J. D. Ferguson, "Variable duration models for speech," in *Proc. Symp. Appl. Hidden Markov Models to Text and Speech*, J. D.Ferguson, Ed., IDA-CRD, Princeton, NJ, 1980.
[26] L. R. Bahl, R. Bakis et al., "Recognition results with several experimental acoustic processors," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Washington, DC, Apr. 1979, pp. 249–251.

Lawrence R. Rabiner (S'62–M'67–SM'75–F'75), for a photograph and biography, see this issue, p. 560.

Stephen E. Levinson (S'72–M'74–SM'83) was born in New York, NY, on September 27, 1944. He received the B.A. degree in engineering sciences form Harvard University, Cambridge, MA, in 1966, and the M.S. and Ph.D. degrees in electrical engineering from the University of Rhode Island, Kingston, in 1972 and 1974, respectively.

From 1966 to 1969, he was a Design Engineer at the Electric Boat Division of General Dynamics, Groton, CT. From 1974 to 1976, he held a J. Willard Gibbs Instructorship in Computer Science at Yale University, New Haven, CT. In 1976, he joined the Technical Staff at Bell Laboratories, Murray Hill, NJ where he is pursuing research in the areas of speech recognition and cybernetics. In 1984 he held a visiting fellowship in the Department of Engineering at Cambridge University, Cambridge, England.

Dr. Levinson is a member of the Association for Computing Machinery and a Fellow of the Acoustical Society of America. He is a member of the editorial board of *Speech Technology*, an Associate Editor of the IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, and Vice-Chairman of the ASSP Technical Committee on Speech.