AT&T BELL LABORATORIES TECHNICAL JOURNAL
Vol. 63, No. 4, April 1984
Printed in U.S.A.

On the Use of Hidden Markov Models for Speaker-Independent Recognition of Isolated Words From a Medium-Size Vocabulary

By L. R. RABINER,* S. E. LEVINSON,* and M. M. SONDHI*

(Manuscript received September 19, 1983)

Recent work at AT&T Bell Laboratories has shown how the theories of Vector Quantization (VQ) and Hidden Markov Modeling (HMM) can be applied to the recognition of isolated word vocabularies. The initial experiments with an HMM word recognizer were restricted to a vocabulary of 10 digits. For this simple vocabulary with dialed-up telephone recordings, we found that a high-performance, speaker-independent word recognizer could be implemented, and that the performance was, for the most part, insensitive to parameters of both the HMM and the VQ. In this paper we extend our investigations of the HMM recognizer to the recognition of isolated words from a medium-size vocabulary (129 words), as used in the AT&T Bell Laboratories airlines reservation and information system. For this moderately complex word vocabulary, we have found that recognition accuracy is indeed a function of the HMM parameters (i.e., the number of states in the model and the number of symbols per state). We have also found that a VQ that includes energy information gives better performance than a conventional VQ of the same size (i.e., same number of code-book entries).

I. INTRODUCTION

Vector Quantizers (VQs), Linear Predictive Coding (LPC) coefficients, and Hidden Markov Models (HMMs) have been shown to be useful for a wide range of speech processing problems in the areas of

* AT&T Bell Laboratories.

coding, synthesis, and recognition.[1-12] In the area of speech recognition, there have been two distinctly different ways of applying HMMs. In the earliest work, extremely ambitious large-scale network models were used to model continuous discourse with constrained vocabularies.[5,7,8] For these large networks, HMMs were derived for basic speech sounds (e.g., phonemes), and words were made by coupling together the individual sound HMMs. Similarly, a sentence was made by coupling the models of all the words in the sentence. Solving such large-scale networks (i.e., finding the best path through the network in order to decode the sentence) was a major problem and required very sophisticated network search routines to find good (generally suboptimal) solutions.

More recently, work has been carried out on speech recognition based on HMMs for individual, isolated words.[11,12] For this type of system, an HMM is designed for each word in the vocabulary, and recognition is carried out by evaluating the probability that an unknown test pattern is the output of a given word model. The reference word whose model has the highest probability is chosen as the recognized word. To date, the word-based HMM recognizer has been tested only on a vocabulary of the 10 digits. For this vocabulary, it was found that the overall performance of the HMM recognizer was fairly insensitive to the parameters of both the HMM and the VQ, and that average word recognition accuracy was approximately the same as that obtained from a conventional Dynamic Time Warping (DTW) template approach.

Based on the success of the word-based HMM recognizer for the digits vocabulary, we have attempted to extend the approach to handle a fairly complex, medium-size word vocabulary. The vocabulary we have chosen is the 129-word airlines terms vocabulary, which has been extensively studied in the context of an airlines reservation and information system.[13-16] Table I shows the words in this vocabulary. We have studied the effects of varying several of the parameters of the HMM and the VQ on recognizer performance. Results indicate that a significant degradation in average word recognition accuracy is introduced by the use of a VQ, even with as many as 256 vectors in the code book. Results also show that the HMM/VQ system can and does achieve average word recognition accuracy comparable to a DTW/VQ template-based recognizer. Finally, we have found that, for a given size of code book, VQs using energy, along with LPC shape, give better performance than VQs with LPC shape alone.

The organization of this paper is as follows. In Section II we briefly review the design of the conventional DTW recognizer, the design of a VQ based on either LPC shape alone or LPC shape plus energy, and the operation of the HMM recognizer. In Section III we describe the

Table I—Words in airlines 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

experimental evaluation used to measure performance of the various recognizers on the 129-word airlines vocabulary. In particular, we give a detailed analysis (for the two best recognition systems) of the individual talker error rates, the most prevalent types of word errors, and the extent to which errors made by DTW and HMM recognizers are disjoint. Finally, in Section IV we summarize our findings.

II. THE DTW AND HMM/VQ WORD RECOGNIZERS

The two types of word recognition systems that we will be concerned with are:

1. A conventional DTW recognizer (either with or without VQ) based on LPC modeling and using isolated word templates as reference patterns.

2. An HMM recognizer with quantized LPC vectors, using single-word Markov models as parametric representations of the words in the vocabulary.

In this section we briefly review the implementations of the VQ, the DTW recognizer, and the HMM recognizer.

2.1 VQ of LPC parameters

Vector quantization is a technique for coding an LPC vector into one of M^* code-book entries such that the average quantization distortion is minimized over some typical training set of LPC vectors. Vector quantization differs from the more conventional scalar quantization methods in that the entire LPC vector is quantized in a single pass, rather than quantizing each component of the vector by a separate quantizer. Experience indicates a substantial savings in required code-book size (bit rate) for VQ over conventional scalar quantization.[1-4] For HMM recognition purposes, VQ serves as a way of characterizing continuous LPC vectors by a set of discrete symbols (i.e., the indices of the code-book entries that provide best matches to the input LPC vectors to the recognizer). In addition, for the DTW recognizer, VQ provides a simple and straightforward way of trading off storage and computation in the calculation of LPC distances as required in the DTW algorithm.[9,12]

A VQ is designed from a training set of I LPC vectors, $\mathbf{a}_i$, $i = 1, 2, \cdots, I$, which are intended to be a good representation of the range of LPC vectors that occur when the words in the vocabulary are pronounced by a wide range of talkers. The VQ training algorithm determines an optimum set of code-book LPC vectors, $\hat{\mathbf{a}}_m$, $m = 1, 2, \cdots, M^*$, such that, for a given M^*, the average distortion in replacing each of the training set vectors, $\mathbf{a}_i$, by the closest code-book entry, $\hat{\mathbf{a}}_m$, is minimum.

If we define $d(\mathbf{a}_R, \mathbf{a}_T)$ as the conventional LPC distance[17] between the LPC vectors $\mathbf{a}_R$ and $\mathbf{a}_T$, i.e.,

$$d(\mathbf{a}_R, \mathbf{a}_T) = \frac{\mathbf{a}_R' V_T \mathbf{a}_R}{\mathbf{a}_T' V_T \mathbf{a}_T} - 1, \tag{1}$$

where V_T is the autocorrelation matrix of the sequence that gave rise to LPC vector $\mathbf{a}_T$, then the goal of the VQ training algorithm is to find the set (of code-book entries), $\hat{\mathbf{a}}_m$, such that

$$\| D_{M^*} \| = \min_{\hat{a}_m} \left\{ \frac{1}{I} \sum_{i=1}^{I} \min_{1 \le m \le M^*} [d(\hat{\mathbf{a}}_m, \mathbf{a}_i)] \right\} \tag{2}$$

is satisfied. The quantity $\|D_{M^*}\|$ is the (minimum) average distortion of the VQ with M^* code-book entries.

The way in which eq. (2) is solved has been intensively investigated by several researchers[1–4] and will not be described here. As a result of these earlier studies, a highly reliable and robust procedure for the design of a VQ code book exists and can readily be implemented.

Although the conventional VQ (which we call a shape VQ) used only the LPC vector, recent studies have shown how normalized frame energy can be incorporated directly into the local distance [eq. (1)][18] to give a shape plus energy VQ. For these designs we denote the LPC distance of eq. (1) as $d_{\text{LPC}}(\mathbf{a}_R, \mathbf{a}_T)$ and we denote an energy distance as $d_E\ (\tilde{E}_R, \hat{E}_T)$. The total distance is then

$$d(\mathbf{a}_R, \mathbf{a}_T) = d_{\text{LPC}}(\mathbf{a}_R, \mathbf{a}_T) + \alpha d_E(\hat{E}_R, \hat{E}_T), \tag{3}$$

where $\hat{E}_R$ and $\hat{E}_T$ are log energies of the reference and test frames, normalized to the peak energies in the word, and α is a multiplier for giving appropriate weight to the energy distance. Using the modified frame distance of eq. (3), a shape plus energy VQ can be designed according to the criteria of eq. (2) with no further modification.

2.2 The conventional DTW word recognizer

Figure 1 shows a block diagram of the conventional DTW word recognizer based on LPC modeling.[17,19] The input speech signal, $s(n)$, recorded over a standard dialed-up telephone line, is bandpass filtered between 100 and 3200 Hz, and digitized at a 6.67-kHz rate. The first step (preprocessing) consists of a first-order digital network, which provides a high-frequency preemphasis to the speech. The preemphasized signal is blocked into frames of 45 ms (300 samples), with each consecutive frame spaced 15 ms (100 samples) apart. An eight-pole

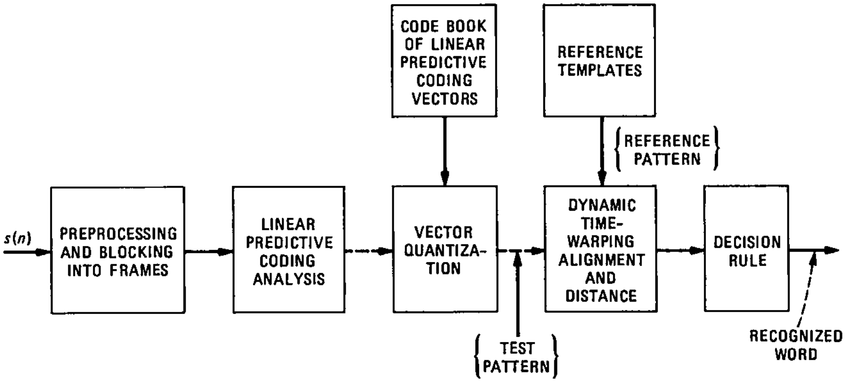
CODE BOOK OF LINEAR PREDICTIVE CODING VECTORS
REFERENCE TEMPLATES
REFERENCE PATTERN
s(n)
PREPROCESSING AND BLOCKING INTO FRAMES
LINEAR PREDICTIVE CODING ANALYSIS
VECTOR QUANTIZATION
DYNAMIC TIME-WARPING ALIGNMENT AND DISTANCE
DECISION RULE
TEST PATTERN
RECOGNIZED WORD

Fig. 1—Block diagram of conventional DTW word recognizer based on LPC modeling.

LPC analysis (autocorrelation method) is performed on each frame of the word (after the word has been isolated by means of an endpoint detector).[20] Each resulting LPC vector is either used directly or vector quantized by means of a code book of size M^*. The resulting sequence of LPC vectors, called the test pattern, is compared with each reference pattern in the reference template set using a DTW alignment algorithm that simultaneously provides a distance score associated with the alignment. The distance scores for all the reference patterns are sent to a decision rule, which provides a classification of the spoken word, and possibly an ordered (by distance) set of the best β candidates.

The word reference patterns for the recognizer of Fig. 1 are created by a training algorithm. For speaker-trained applications, typically a single reference pattern is created for each word in the vocabulary using a robust training algorithm.[21] For speaker-independent applications, a set of Q reference patterns is created for each vocabulary word using a clustering procedure.[22,23] Typically, about 12 templates per word are sufficient for recognizing words from a fairly homogeneous adult population of native American talkers.

If a VQ code book of M^* entries has been designed as in Section 2.1, one can compute and store the table of $M^* \times M^*$ distances between all pairs of code-book entries. In this manner, computation of distance between any pair of VQ code-book entries becomes a simple table lookup. Hence, if we vector quantize the LPC vectors of a test utterance and of all reference patterns, then the computation for distances in the DTW matching becomes trivial. In addition, a novel technique for reducing quantization distortion when using a VQ was proposed by Sakoe.[24] For this technique the test vector is not quantized. Instead, the table of distances between each test vector and all code-book entries is computed once and used in the DTW distance calculation. In this manner there is reduced distortion, reduced storage (over conventional VQ), and essentially no computation for local distance calculation (it is still a table lookup procedure).

2.3 The HMM word recognizer

Figure 2 is a block diagram of the HMM word recognizer. The front-end processing, namely preprocessing, frame blocking, LPC analysis, and vector quantization, is identical to that used in the vector quantized DTW recognizer described above. The test utterance is reduced to an observation sequence, $\{\mathbf{O}\}$, consisting of the indices of the code-book vectors that best match corresponding LPC vectors of the utterance. A Viterbi scoring algorithm determines, for each individual word HMM, the probability that the observation sequence was generated by the given word HMM. A decision rule either chooses the word

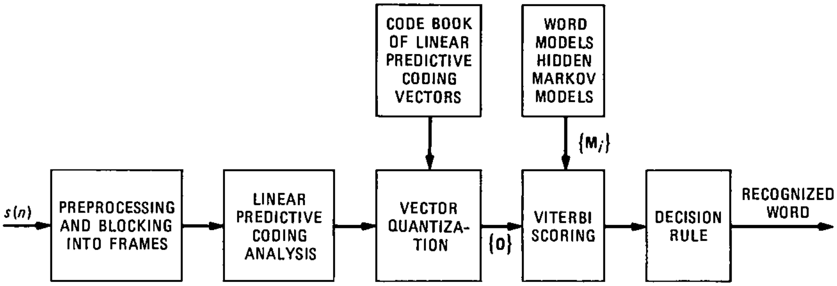
CODE BOOK OF LINEAR PREDICTIVE CODING VECTORS
WORD MODELS HIDDEN MARKOV MODELS
{M_i}
s(n)
PREPROCESSING AND BLOCKING INTO FRAMES
LINEAR PREDICTIVE CODING ANALYSIS
VECTOR QUANTIZATION
{O}
VITERBI SCORING
DECISION RULE
RECOGNIZED WORD

Fig. 2—Block diagram of HMM word recognizer based on LPC modeling and VQ.

whose word model has the highest probability as the recognized word, or gives a list of word candidates ordered by model probability scores.

Model probability scores for each word model are computed as follows. Each word HMM is an N-state model characterized by a state transition matrix, **A**, and a model output symbol probability matrix, **B**. Figure 3 illustrates an $N = 5$ state word model with M^* discrete output symbols for each state. Here we assume that the word models are left-to-right models; i.e., the elements of the transition matrix, a_{ij}, satisfy the relationship

$$a_{ij} = 0 \qquad j < i \tag{4a}$$

and, furthermore, we restrict the range of transitions to the case

$$a_{ij} = 0 \qquad j > i + 2. \tag{4b}$$

That is, we allow transitions between states that are either adjacent or one apart. Previous experimentation has shown these constraints are reasonable.[12]

Based on the above discussion, the scoring procedure for the observation sequence $\mathbf{O} = \{O_1, O_2, \cdots, O_L\}$ (i.e., L indices of code-book entries), given the model $\mathbf{M}(\equiv \mathbf{A}, \mathbf{B})$, is as follows:

1. Initialization: $\delta_1(1) = \log[b_1(O_1)]$

   $$\delta_1(i) = \infty, 2 \leqslant i \leqslant N$$

2. Recursion: For $2 \leqslant l \leqslant L$, $1 \leqslant j \leqslant N$

   $$\delta_l\ (j) = \max\{\delta_l\ -1(i) + \log[a_{ij}]\} + \log[b_j(O_l)]$$

   $$\min(1, j - 2) \leq i \leq \max(j, N)$$

3. Termination: $P(\mathbf{M}) = \delta_L(N)$.

The above algorithm is a form of the well-known dynamic programming method and can be shown to have the property of determining the state sequence $\mathbf{i} = i_1, i_2, \cdots, i_L$, which maximizes $P(\mathbf{O},\mathrm{i} \mid \mathbf{M})$. It can be seen that if the entries in the matrices **A** and **B** are stored in

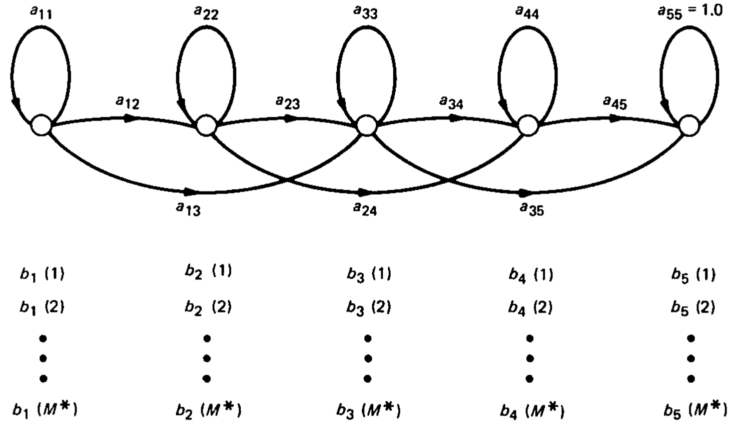

Fig. 3—Typical five-state Markov model for word with transition matrix $\mathbf{A} = \{a_{ij}\}$ and output symbol matrix $\mathbf{B} = \{b_j(k)\}$.

logarithmic format, i.e., $\log[a_{ij}]$, and $\log[b_j(k)]$, then the computation of the Viterbi scoring algorithm requires no multiplications or logarithm computation. Hence, the speed of computation of the Viterbi scoring is quite high.

The **A** and **B** matrices for each word HMM are estimated from a training set consisting of a set of observation sequences for the word. Starting with an initial estimate of the model, the probability, P, of observing the given training sequence **O** given the model **M** is computed. Using the Baum-Welch reestimation algorithm,[25] the model is iteratively adjusted to increase P. The iterations are stopped when P stops increasing significantly or when some other stopping criterion is met (e.g., when the number of iterations exceeds a limit).

III. RECOGNITION EVALUATION OF THE AIRLINES VOCABULARY

A series of recognition tests was run on both the DTW and HMM word recognizers, using the 129-word airlines vocabulary of Table I. A training set consisting of one replication of each word by each of 100 talkers (50 male, 50 female) was used to generate speaker-independent word templates for the DTW recognizer, and speaker-independent word HMMs for the HMM recognizer. The code-book vectors for the VQ were also derived from a subset of the training data. (VQs derived from a digits-only vocabulary were also tried and produced essentially no degradation in system performance.) All word recordings were made over dialed-up local telephone lines, with a new line used for each talker.

An independent test set consisting of one replication of each word by each of 20 talkers (10 male, 10 female) was used. None of the test talkers had contributed to the training data. Again, all recordings were made over local, dialed-up telephone lines.

A series of 14 separate recognition tests were made on the DTW and HMM recognizers. The runs consisted of the following systems:

Run 1: Conventional DTW recognizer without VQ.

Runs 2 and 3: Conventional DTW recognizer with VQ applied to both test and reference sets.

Runs 4 through 9: HMM recognizer using a shape VQ with different values for N, the number of states, and M^*, the size of the VQ code book.

Run 10: HMM recognizer using a shape plus energy VQ.

Run 11: HMM recognizer using a shape VQ with variable number of states per word.

Run 12: HMM recognizer using a shape VQ with five randomly generated models per word.

Run 13: HMM recognizer using a shape VQ using the average of five randomly generated models per word.

Run 14: HMM recognizer using a shape VQ with both five and eight state models for each word.

Recognition runs 1 through 10 are the standard cases of DTW and HMM recognizers, and they effectively provide performance measurements on the effects of HMM and VQ parameters. Run 11 is used to test the hypothesis that long words (in terms of phoneme count) need bigger models (more states) than short words. For this run the number of states in the Markov model was set equal to the number of phonemes in the word, and a lower limit of 4 states and an upper limit of 10 states were imposed. Runs 12 and 13 examine the effects of generating multiple models (from different random initial model guess) for each word and either using all models in the recognizer (run 12), or the averge model (run 13). Finally, run 14 is used to study the effects of using multiple models with different numbers of states in each model. For this run, models with five and eight states were generated for each word.

3.1 Recognition run results

The performance results, given as a series of average word error rates for the β best word candidates, are shown in Table II and are partially plotted in Figs. 4 and 5. Table II gives the average word error rate scores for each of the 14 runs. Figure 4 shows plots of these results for runs 1, 2, 3, and 10, and Fig. 5 shows plots of the results for runs 4 through 9. An examination of the curves in Fig. 4 shows that the use of the VQ leads to significantly poorer performance in both the DTW and the HMM recognizers (at least 6 percent for $\beta = 1$ and at

Table II—Characteristics of recognizer and average word error rate scores (%) as function of candidate position for individual recognition tests on airlines vocabulary

				Number of Word Candidates				
Run	Recognizer	*N*	*M**	1	2	3	4	5
1	DTW	—	—	9.4	4.0	2.4	1.7	1.4
2	DTW/VQ	—	64	20.1	10.7	6.0	4.2	3.5
3	DTW/VQ	—	128	16.4	7.4	4.5	3.6	2.8
4	HMM/VQ	5	64	27.2	15.6	11.1	8.4	6.7
5	HMM/VQ	5	128	22.5	12.3	8.2	6.1	4.8
6	HMM/VQ	8	64	24.0	13.3	8.8	6.6	5.3
7	HMM/VQ	8	128	20.7	10.3	6.4	4.8	3.9
8	HMM/VQ	10	128	19.3	9.9	6.2	4.3	3.6
9	HMM/VQ	10	256	17.6	8.2	5.1	3.8	2.9
10	HMM/VQ.E	10	128	15.5	7.0	4.4	3.1	2.2
11	HMM/VQ	Variable	64	30.5	19.1	13.4	9.9	8.1
12	$(\text{HMM/VQ})_5$	5	64	30.4	18.4	11.9	9.1	7.3
13	$\overline{(\text{HMM/VQ})_5}$	5	64	30.4	18.4	11.9	9.1	7.3
14	HMM/VQ	5 + 8	64	24.3	13.5	8.9	6.7	5.5

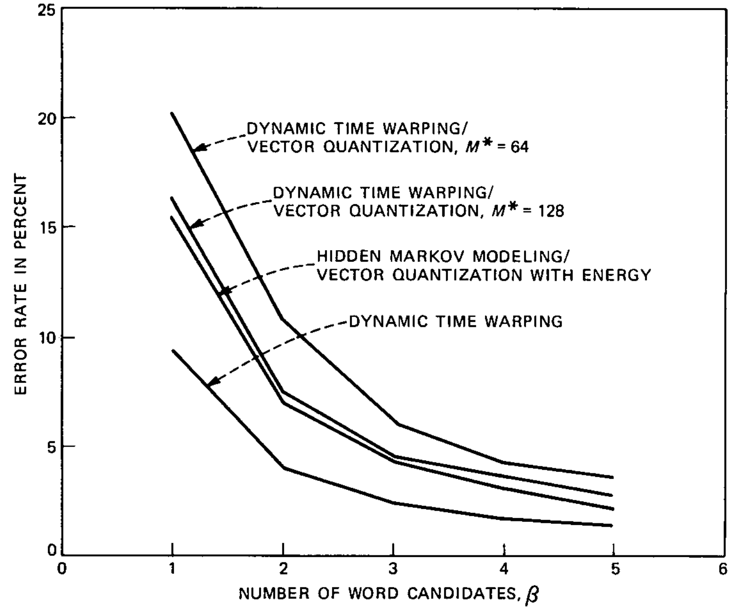

Fig. 4—Average word error rate, in percent, versus number of word candidates for data of runs 1 through 3 and run 10.

least 0.8 percent for $\beta = 5$). However, within the set of recognizers using a VQ, the HMM recognizer with a shape plus energy VQ achieves performance comparable to or better than the DTW recognizer with a shape VQ of the same size.

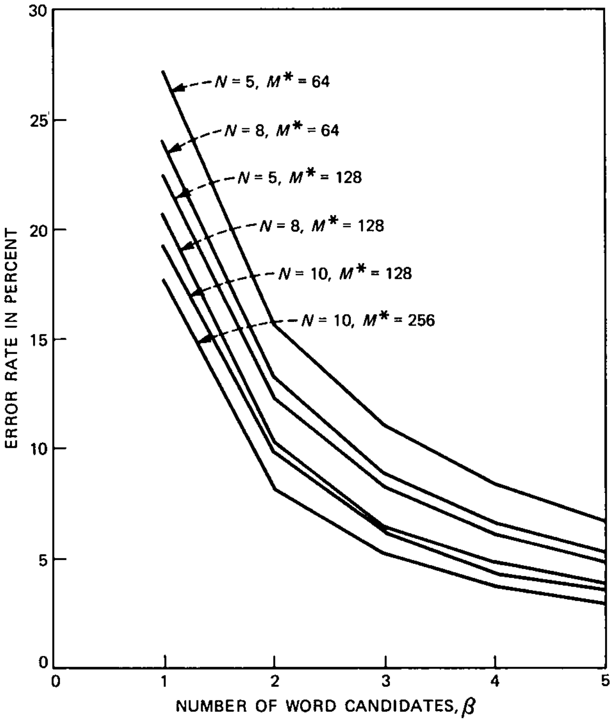

Fig. 5—Average word error rate, in percent, versus number of word candidates for data of runs 4 through 9.

The results in Fig. 5 show that the parameters N and M^* of the HMM (and the VQ) do affect recognizer performance significantly for the airlines vocabulary. The results shown in this figure indicate that increasing both N and M^* leads to improved word recognition accuracy. In fact, a decrease in average word error rate of about 10 percent is obtained in going from $N = 5$, $M^* = 64$ to $N = 10$, $M^* = 256$.

The results of the special runs (11 through 14) have some interesting implications. The results of run 11 (with variable-size Markov models) indicate worse performance than for fixed-size models. This result was anticipated based on experimentation with simulated HMM examples in which a stronger bias was always found for bigger models (more states) than for smaller models. The results of runs 12 and 13, in which five random starting conditions were used to generate five models per word, show a small but consistent improvement in performance when using all five models in the scoring (run 12) and a small but consistent degradation in performance when using the average model in the scoring (run 13). A plausible explanation of this result is that due to large model variability, averaging often results in a poor model because of the outliers.

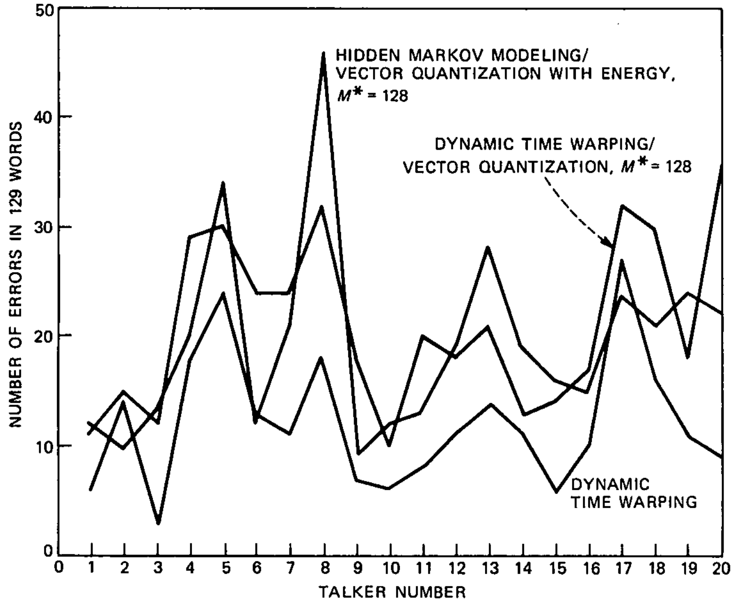

Fig. 6—Number of word errors as function of talker for data of runs 1, 3, and 10.

The results of run 14, in which the five- and eight-state models were combined, show that the performance in this case is essentially identical to the eight-state recognizer alone. Hence, this result again supports the observation that a strong bias exists in favor of bigger models.

3.2 Additional analysis of runs 1, 3, and 10 data

For the data of runs 1, 3, and 10, several additional analyses were performed to determine the correlation of individual talker errors with the recognizer, to examine the most frequent errors made, and to see how similar the errors were in the two recognizers. Figure 6 shows a plot of the number of word errors versus talker number for the recognition systems of runs 1, 3, and 10. It can be seen that there is a high correlation in the individual talker error rates among the three systems. It can also be seen that there is a high degree of variability in individual talker error rates.

Table III shows, for the data of runs 1 and 10, the words with the highest error rates over all talkers, and the words most confused with these words. The numbers in parentheses indicate the total number of errors (across the 20 talkers) and the total number of confusions. For the DTW recognizer the confusions of these 12 words accounted for one-third of the overall word errors; for the HMM recognizer the confusions among the 14 worst words accounted for 30 percent of the

Table III—Major word confusions in data of runs 1 and 10

DTW Recognizer		HMM Recognizer	
Word	Major Confusions	Word	Major Confusions
By (9)	I (5), My (4)	Many (11)	May (6)
Do (8)	To/Two (4)	Oh (10)	Of (3), How (2)
Flight (8)	Flights (6), Like (2)	Time (10)	Times (5), Five (4)
May (7)	Make (2)	Pay (10)	A (5), Take (3)
Oh (7)	How (3), Go (2)	Flight (9)	Flights (6)
Is (6)	In (5)	Do (8)	To/Two (8)
Home (6)	Oh (2), How (2)	On (8)	Of (3), August (2)
Seat (6)	Seats (3)	Please (8)	Leave (4)
Time (6)	Times (4)	Miami (8)	Monday (4), Morning (2)
Some (6)	From (3)	In (7)	Evening (3), A.M. (2)
A (6)	Take (2)	Want (7)	What (4), One (2)
Seats (6)	Seat (4)	Take (7)	Eight (3), Seat (2)
		Seats (7)	Seat (5)
		Three (7)	Three (2), Many (2), May (2)

overall word errors. For the DTW recognizer most of the major confusions are between words that sound similar (e.g., "By," "I," "My"); for the HMM recognizer there are many major confusions between dissimilar words (e.g., "In," "Evening;" "Miami," "Morning," etc.). It is interesting that among the words most confused by both systems, there are only three overlapping words, namely "Do," "Time," and "Seats."

The final check on the data was a comparison of the errors made by the DTW recognizer (run 1) and the HMM recognizer (run 10). In earlier experimentation with the digits vocabulary, it was found that an orthogonality existed between the errors of the DTW and HMM recognizers, in that whenever the DTW recognizer made an error, the HMM recognizer was almost always correct. We have made the same type of orthogonality check on the data of runs 1 and 10. Of the 243 word errors made by the DTW recognizer, 111 (45 percent) were cases in which the HMM recognizer also made an error. Hence, in only 55 percent of the cases is there any potential for error detection and correction. Thus, we conclude that it would not be easy to combine the results of the DTW and HMM recognizers, for this vocabulary, to greatly improve overall word accuracy.

IV. SUMMARY

In this paper we have applied isolated word recognition based on probabilistic modeling (HMMs) to a medium-size vocabulary of airline terms. We have found that the use of a finite-size VQ leads to a significant degradation in performance for both the conventional DTW recognizer and the one based on HMMs. We have also found that after vector quantization the HMM and DTW recognizers can be designed to give essentially identical performance. We have shown that increasing N, the number of states in the HMM, and/or M^*, the size of the VQ code book, improves performance of the HMM recognizer. Hence, one should use the largest models that can be accommodated in a practical implementation. We have also found that using a VQ based on both LPC shape and energy gives improved recognition performance over a VQ based on LPC shape alone. Hence, the energy contour is extremely helpful in recognizing word vocabularies with many polysyllabic words.

The only discouraging finding was that there was a high degree of overlap between the word errors made by the DTW and HMM recognizers. Hence, there appears to be no simple method for combining these two approaches to give greatly improved performance.

REFERENCES

1. Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantization," IEEE Trans. Commun., *COM-28,* No. 1 (January 1980), pp. 84–95.
2. B. Juang, D. Wong, and A. H. Gray, Jr., "Distortion Performance of Vector Quantization for LPC Voice Coding," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-30,* No. 2 (April 1982), pp. 294–303.
3. A. Buzo et al., "Speech Coding Based Upon Vector Quantization," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-28,* No. 5 (October 1980), pp. 562–74.
4. D. Wong, B. Juang, and A. H. Gray, Jr., "An 800 Bit/S Vector Quantization LPC Vocoder," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-30,* No. 5 (October 1982), pp. 770–80.
5. J. K. Baker, "The DRAGON System—An Overview," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-23,* No. 1 (February 1975), pp. 24–9.
6. A. B. Poritz, "Linear Predictive Hidden Markov Models and the Speech Signal," Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Paris, France (May 1982), pp. 1291–4.
7. F. Jelinek, L. R. Bahl, and R. L. Mercer, "Design of a Linguistic Decoder for the Recognition of Continuous Speech," IEEE Trans. Inform. Theory, *IT-21* (May 1975), pp. 250–6.
8. L. R. Bahl, F. Jelinek, and R. L. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition," IEEE Trans. Pattern Analysis and Machine Intelligence, *PAMI-5,* No. 2 (March 1983), pp. 179–90.
9. A. Buzo, H. Martinez, and C. Rivera, "Discrete Utterance Recognition Based Upon Source Coding Techniques," Proc. ICASSP-82 (May 1982), pp. 539–42.
10. J. E. Shore and D. Burton, "Discrete Utterance Speech Recognition Without Time Normalization," Proc. ICASSP-82 (May 1982), pp. 907–10.
11. R. Billi, "Vector Quantization and Markov Source Models Applied to Speech Recognition," Proc. ICASSP-82 (May 1982), pp. 574–7.
12. 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," B.S.T.J., *62,* No. 4 (April 1983), pp. 1075–105.
13. A. E. Rosenberg and F. Itakura, "Evaluation of an Automatic Word Recognition System Over Dialed-Up Telephone Lines," J. Acoust. Soc. Amer., Supplement 1, (1976), p. 60.
14. S. E. Levinson, A. E. Rosenberg, and J. L. Flanagan, "Evaluation of a Word Recognition System Using Syntax Analysis," B.S.T.J., *57,* No. 5 (May–June 1978), pp. 1619–26.
15. S. E. Levinson and A. E. Rosenberg, "A New System For Continuous Speech Recognition—Preliminary Results," Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Washington, DC (April 1979), p. 239–43.
16. S. E. Levinson and K. L. Shipley, "A Conversational Mode Airline Information and Reservation System Using Speech Input and Output," B.S.T.J., *59,* No. 1 (January 1980), pp. 119–37.

17. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-23,* No. 1 (February 1975), pp. 67–72.
18. 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., *63,* No. 5 (May–June 1984).
19. L. R. Rabiner and S. E. Levinson, "Isolated and Connected Word Recognition—Theory and Selected Applications," IEEE Trans. Commun., *COM-29,* No. 5 (May 1981), pp. 621–59.
20. L. F. Lamel et al., "An Improved Endpoint Detector for Isolated Word Recognition," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-29,* No. 4 (August 1981), pp. 777–85.
21. L. R. Rabiner and J. G. Wilpon, "A Simplified Robust Training Procedure for Speaker Trained, Isolated Word Recognition Systems," J. Acoust. Soc. Amer., *68,* No. 5 (November 1980), pp. 1271–6.
22. S. E. Levinson et al., "Iterative Clustering Techniques for Selecting Speaker Independent Reference Templates for Isolated Word Recognition," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-27,* No. 2 (April 1979), pp. 134–41.
23. L. R. Rabiner et al., "Speaker Independent Recognition of Isolated Words Using Clustering Techniques," IEEE Trans. Acoustics, Speech, and Signal Processing, *ASSP-27,* No. 4 (August 1979), pp. 336–49.
24. H. Sakoe, "Device for Recognizing an Input Pattern With Approximate Patterns Used for Reference Patterns on Mapping," U.S. Patent 4,256,924, March 17, 1981.
25. L. E. Baum et al., "A Maximization Techique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," Ann. Math. Statist., *41,* No. 1 (1970), pp. 164–71.

AUTHORS

Stephen E. Levinson, B.A. (Engineering Sciences), 1966, Harvard University; M.S and Ph.D (Electrical Engineering), University of Rhode Island, Kingston, 1972 and 1974, respectively; AT&T Bell Laboratories, 1976—. From 1966 to 1969, Mr. Levinson was a design engineer at Electric Boat Division of General Dynamics in Groton, Connecticut. From 1974 to 1976, he held a J. Willard Gibbs Instructorship in Computer Science at Yale University. As a Member of Technical Staff at AT&T Bell Laboratories, he is pursuing research in the areas of speech recognition and cybernetics. Fellow, Acoustical Society of America; Senior Member, IEEE; Member, Association for Computing Machinery. Mr. Levinson is also a member of the editorial board of Speech Technology and an associate editor of the IEEE Transactions on Acoustics, Speech, and Signal Processing.

Lawrence R. Rabiner, S.B. and S.M., 1964, Ph.D (Electrical Engineering), 1967, Massachusetts Institute of Technology; AT&T Bell Laboratories, 1962—. At AT&T Bell Laboratories Mr. Rabiner is engaged in research on speech communications and digital signal processing techniques. He is coauthor of the books *Theory and Application of Digital Signal Processing* (Prentice-Hall, 1975), *Digital Processing of Speech Signals* (Prentice-Hall, 1978), and *Multirate Digital Signal Processing* (Prentice-Hall, 1983). Fellow, Acoustical Society of America, IEEE; Member, Eta Kappa Nu, Sigma Xi, Tau Beta Pi, the National Academy of Engineering.

Man Mohan Sondhi, B.Sc. (Physics), Honours, 1950, Delhi University, Delhi, India; D.I.I.Sc. (Communications Engineering), 1953, Indian Institute of Science, Bangalore, India; M.S. and Ph.D (Electrical Engineering), University of Wisconsin, Madison, Wisconsin, 1955 and 1957, respectively; AT&T

Bell Laboratories, 1962—. Before joining AT&T Bell Laboratories, Mr. Sondhi worked at the Avionics Division of John Oster Manufacturing Co., Racine, Wisconsin and the Central Electronics Research Institute in Pilani, India. He taught for one year at Toronto University, Toronto, Canada. At AT&T Bell Laboratories his research has included work on speech signal processing; echo cancellation; adaptive filtering; modeling of auditory, speech and visual processing by human beings; acoustical inverse problems; and, more recently, speech recognition based on hidden Markov modeling of speech. From 1971 to 1972 Mr. Sondhi was a guest scientist at the Royal Institute of Technology, Stockholm, Sweden.