Open access • Proceedings Article • DOI:10.1109/ICASSP.1985.1168456

Discrete utterance speech recognition without time alignment — Source link

K.L. Brown, V.R. Algazi

Institutions: University of California, Davis

Published on: 26 Apr 1985 - International Conference on Acoustics, Speech, and Signal Processing

Topics: Speech corpus, Speech processing, Speech analytics, Audio mining and Speech technology

Discrete Utterance Speech Recognition Without Time Alignment

JOHN E. SHORE, SENIOR MEMBER, IEEE, AND DAVID K. BURTON, MEMBER, IEEE

Abstract—**The results of a new method are presented for discrete utterance speech recognition. The method is based on rate-distortion speech coding (speech coding by vector quantization), minimum cross-entropy pattern classification, and information-theoretic spectral distortion measures. Separate vector quantization code books are designed from training sequences for each word in the recognition vocabulary. Inputs from outside the training sequence are classified by performing vector quantization and finding the code book that achieves the lowest average distortion per speech frame. The new method obviates time alignment. It achieves 99 percent accuracy for speaker-dependent recognition of a 20-word vocabulary that includes the ten digits, with higher accuracy for recognition of the digit subset. For speaker-independent recognition, the method achieves 88 percent accuracy for the 20-word vocabulary and 95 percent for the digit subset. Background of the method, detailed empirical results, and an analysis of computational requirements are presented.**

I. Introduction

CURRENTLY successful approaches to discrete utterance speech recognition involve time alignment [1], [2]. From an unknown input utterance, a feature vector is obtained every 10–30 ms by making a set of measurements. The resulting sequence of feature vectors is classified by comparing it to a set of prestored reference sequences derived from training data and finding the best match. An important step in these comparisons is the alignment of the input sequence in time with each reference sequence. In the simplest form of alignment, the endpoints of the input sequence are aligned with the endpoints of the reference sequence and the intervening input data is stretched or compressed linearly in time. Because variations in speaking rates are nonlinear, the best current systems use more sophisticated methods in which the time axis of the input data is transformed by a constrained, nonlinear warping function that is chosen by dynamic programming to provide the best possible match between the input sequence and the reference sequence [3], [4], [5]. Such methods generally are called dynamic time warping (DTW).

The time-alignment approach is successful—a recent test of commercial speech recognizers yielded accuracies in the 90 percent to 99 plus percent range for a 20-word vocabulary [2]. The approach is also intuitively compelling—it seems obvious that accurate recognition requires the exploitation of time sequence information.

It appears, however, that time sequence information is less critical than is commonly assumed. In this paper we present recent results of a new method for discrete utterance speech recognition. The new method does not use time alignment—indeed, no time sequence information is used at all. Nevertheless, the method achieves 99 percent accuracy for the same 20-word vocabulary used in [2]. The method is based on a variety of ideas and methods from information theory and related fields, namely, rate-distortion speech coding (speech coding by vector quantization) [6], [7], minimum cross-entropy pattern classification [8], and information-theoretic spectral distortion measures [9]. We reported initial results in [10], [11]. Similar work has been reported by authors from Japan [12], Mexico [13], and the United States (Bell Laboratories) [14].

II. Background

Speech coding by vector quantization [6], [7] is a narrow-bandwidth speech coding technique based on linear predictive coding (LPC). Input speech is divided into a sequence of fixed-length segments called frames—typical frame lengths are 20–30 ms. Using estimates of the sample autocorrelation function that are measured in each frame, the shape of the speech spectrum in each frame is coded in terms of the identity of a prestored set of LPC parameters that defines an autoregressive model and is called a *codeword*. The parameters used are the inverse filter gain squared σ^2 and sample coefficients a_i, $i = 1, \cdots, M$, with $a_0 = 1$. The collection of possible codewords is called a *code book*. Let $C = \{C_1, C_2, \cdots, C_N\}$ be a code book of N codewords C_i, each defining an autoregressive model. Let S_j be the autocorrelation estimates from the jth frame of the speech to be coded. Then the shape of the spectrum for the jth frame is coded by identifying the codeword C_b that "best represents" S_j according to the "nearest-neighbor

Editor's Note: One of the duties of the Editor is to serve as Associate Editor for Applications and Miscellaneous Topics. The applications area has always been a difficult one for the TRANSACTIONS: many readers would like to see more applications of information theory in the TRANSACTIONS, but it is not always clear when a paper fits such a description. In addition, many applications oriented papers often tend to drift to other, more applications oriented TRANSACTIONS. On seeing an early draft of this paper, I found it to meet the qualifications of an applications paper admirably, and I invited the authors to publish it in this TRANSACTIONS. It is an empirical study of an information theoretic classification technique applied to the important area of speech recognition. It combines information measures, pattern recognition, and source coding techniques that have appeared among our pages to develop a novel approach to isolated word recognition.

Manuscript received December 22, 1982; revised March 31, 1983.

The authors are with the Computer Science and Systems Branch, Information Technology Division, Naval Research Laboratory, Washington, DC 20375.

rule"

$$d(S_j, C_b) = \min_i d(S_j, C_i), \tag{1}$$

for some distortion measure d. The distortion measure used in [6], [7] is the Itakura–Saito distortion [15], [9].

Speech coding by vector quantization using the Itakura-Saito distortion has strong connections with information theory. In particular, under suitable assumptions the Itakura–Saito distortion can be shown to be a special case of the asymptotic cross-entropy rate between two stochastic processes [9], [7], [16]. Cross-entropy—also called discrimination information, directed divergence, Kullback–Leibler number, etc.—is a measure of information dissimilarity [17], [18]. Cross-entropy minimization, which can be viewed as a general method of inference about probability distributions [19], is useful in a variety of applications [8], [16], [20]–[23]. Speech coding by vector quantization is a particular example [6]. Not only can it be derived directly by cross-entropy minimization [7], it can be derived as a special case of a more general minimum cross-entropy classification method [8]. Specifically, vector quantization using the Itakura–Saito distortion measure is equivalent to choosing the codeword C_b such that the set of input speech parameters S_j provides the least additional information beyond what C_b provides [8].

Vector quantization code books are designed to minimize the average distortion that results from encoding a long training sequence of speech frames. In particular, if $T_j, j = 1, \cdots, L$ is such a training sequence, the code book $\boldsymbol{C}$ is designed so that

$$\frac{1}{L}\sum_{j=1}^{L} d\left(T_j, C_b^{(j)}\right) \tag{2}$$

achieves at least a local minimum, where $C_b^{(j)}$ is the codeword resulting from encoding T_j. If the training sequence comprises a representative sample of the speech to be coded, then $\boldsymbol{C}$ should also encode speech from outside the training sequence with a similarly small distortion. In practice, code books are designed by an iterative, clustering technique. The original ideas were published in [24]; for details about the vector quantizer design algorithm, see [6], [7], [25]. Put simply, the algorithm divides the L frames of the training sequence into N clusters of frames such that all of the frames in any particular cluster have similar spectrum shapes. The N codewords are the centroids of these clusters. Usually, the size of the code book $\boldsymbol{C}$ is a power of 2—i.e., $N = 2^R$. The code book is then known as a *rate R* code book because R bits must be transmitted to identify the best codeword for each speech frame.

III. Description of Approach

In speech coding by vector quantization, a single code book is designed from a training sequence that is as long as computational constraints permit and that is chosen to be representative of all speech to be encoded by the system. In our approach to discrete utterance recognition, we use vocabulary—and we design each code book using a relatively short training sequence containing repetitions of one word in the recognition vocabulary. For example, a code book for the word SEVEN is designed by running the vector quantizer design algorithm on a training sequence of several repetitions of the word SEVEN. When an unknown word is to be classified, every frame of the word is encoded as in (1) using each code book. The average distortion over all frames of the input is computed for each code book, and the input is classified as the word corresponding to the code book yielding the lowest average distortion. A threshold on the average distortion can be used to reject input words that are not in the recognition vocabulary. Note that the basic approach can be used for both single-speaker and multiple-speaker training sequences.

To be more precise, let V be the number of words in the recognition vocabulary. Then there are V code books $\boldsymbol{C}_k$, $k = 1, \cdots, V$—one for each word—which together comprise a *code book set*. Let C_{ki}, $i = 1, \cdots, N_k$, be codewords in $\boldsymbol{C}_k$, where N_k is the size of $\boldsymbol{C}_k$. Let there be L frames of speech in the utterance to be classified, and let S_j be the set of autocorrelation estimates from the jth frame ($j = 1, \cdots, L$). Finally, let D_k be the *average distortion* resulting from coding the utterance with the kth code book,

$$D_k = \frac{1}{L}\sum_{j=1}^{L} d\left(S_j, C_{kb}^{(j)}\right), \tag{3}$$

where $C_{kb}^{(j)}$ is the codeword resulting from encoding S_j with code book $\boldsymbol{C}_k$ as in (1). Then the utterance is classified as the rth word in the recognition vocabulary, where

$$D_r = \min_k D_k. \tag{4}$$

Note that this procedure classifies the unknown utterance without performing any kind of time alignment or normalization.

A problem can occur if one word in the recognition vocabulary is contained within another word. This would happen, for example, if the words SEVEN and SEVENTEEN were both in the recognition vocabulary. In this case, it is likely that an utterance of SEVEN would result in low average distortions from two code books. One way to deal with this is to include in the classification decision the extent to which each code book is "spanned" by the input utterance—if the correct code book is $\boldsymbol{C}_m$, then we expect that b in $C_{kb}^{(j)}$ from (3) will vary over more of $\boldsymbol{C}_m$ than over $\boldsymbol{C}_j$, $j \neq m$. To test this idea, we tried classifying by minimizing D_k/A_k instead of D_k in (4), where A_k is the fraction of the codewords in code book $\boldsymbol{C}_k$ that were selected during the classification of the input utterance. We refer to A_k as the *span fraction*.

If desired, one can select a set of threshold values $D_r^\dagger$ and require $D_r < D_r^\dagger$ in (4) for a valid classification. This can improve classification reliability and can also be useful in rejecting words outside of the recognition vocabulary.

A. Distortion Measures

We used several alternatives for the distortion measure d

the *Itakura–Saito distortion* between them is

$$d_{IS}(f, \hat{f}) = \int_{-\pi}^{\pi} \frac{d\vartheta}{2\pi}\left[\frac{f}{\hat{f}} - \ln\frac{f}{\hat{f}} - 1\right]. \quad (5)$$

As we mentioned earlier, the theoretical significance of d_{IS} arises from maximum-likelihood classification [15] and also from d_{IS} being the asymptotic cross-entropy between the stochastic processes underlying f and $\hat{f}$ [9], [7], [16].

We are concerned with power spectrum estimates f and $\hat{f}$ that have the autoregressive (LPC) form

$$f(\vartheta) = \frac{\sigma^2}{|A(z)|^2}, \quad (6)$$

where

$$A(z) = \sum_{k=0}^{M} a_k z^{-k}$$

and $z = \exp(i\vartheta)$. In this case, (5) can be expressed as [9]

$$d_{IS}(f, \hat{f}) = \frac{\alpha}{\hat{\sigma}^2} + \ln(\hat{\sigma}^2) - \ln(\sigma^2) - 1, \quad (7)$$

where

$$\alpha = r(0)\hat{r}_a(0) + 2\sum_{n=1}^{M} r(n)\hat{r}_a(n),$$

$$\hat{r}_a(n) = \sum_{i=0}^{M-n} \hat{a}_i\hat{a}_{i+n},$$

and where $r(n)$ are the time-domain autocorrelations of $f(\vartheta)$.

The Itakura–Saito distortion between a power spectrum and a scaled version of itself is

$$d_{IS}(f, \lambda f) = \frac{1}{\lambda} + \ln\lambda - 1,$$

which shows that the use of d_{IS} in (1)–(3) could lead to problems if an overall amplifier gain can vary during training and classification. To avoid such problems, two gain-insensitive versions of d_{IS} have been introduced. The first is called the *gain-optimized* Itakura-Saito distortion [9],

$$d_{GO}(f, \hat{f}) \equiv \min_{\lambda>0} d_{IS}(f, \lambda\hat{f}) = \ln\int_{-\pi}^{\pi} \frac{d\vartheta}{2\pi}\left[\frac{f}{\hat{f}}\right] - \int_{-\pi}^{\pi} \frac{d\vartheta}{2\pi}\ln\left[\frac{f}{\hat{f}}\right]. \quad (8)$$

In the case of LPC forms (6), d_{GO} can be expressed as

$$d_{GO}(f, \hat{f}) = \ln(\alpha) - \ln(\sigma^2). \quad (9)$$

The second gain-insensitive measure, the *gain-normalized* Itakura-Saito distortion [9] is defined for spectra of the LPC form (6):

$$d_{GN}(f, \hat{f}) \equiv d_{IS}\left(\frac{f}{\sigma^2}, \frac{\hat{f}}{\hat{\sigma}^2}\right) = \frac{\alpha}{\sigma^2} - 1. \quad (10)$$

For power spectra that have the LPC form (6), the following relationships hold among the three foregoing distortion measures:

$$d_{GO} = \ln(1 + d_{GN})$$

$$d_{GO} = \ln\left[\frac{\hat{\sigma}^2}{\sigma^2}\left(d_{IS} + \ln\frac{\sigma^2}{\hat{\sigma}^2} + 1\right)\right].$$

During classification, the input speech frames provide the argument f in (7), (9), or (10).

B. Code Book Generation

Each classification code book $\boldsymbol{C}_k$ is designed from a separate training sequence containing repetitions of the kth word in the recognition vocabulary. For speaker-dependent experiments, the training sequence for each code book is spoken by one person and the code books are used to classify additional utterances from the same person. For multiple-speaker and speaker-independent experiments, the training sequence for each code book is spoken by several people and the code books are used to classify additional utterances from these people or utterances from different people.

We used two different types of classification code books. The first, called *clustered* code books, are full-search, optimal, vector quantization code books mentioned in Section III and fully described in [6], [7]. Our clustered code book sets were generated either to fixed-size or to fixed-distortion criteria. As the name implies, in a *fixed-size* code book, the size N_k is specified ahead of time and the design algorithm chooses N_k codewords that minimize the average distortion resulting from encoding the training sequence. All code books in a fixed-size code book set have the same size. For a *fixed-distortion* code book, the design algorithm increases the code book size until it can design a code book that encodes the training sequence with an average distortion that is less than or equal to a prespecified value. All code books in a fixed-distortion code book set are generated with the same average distortion threshold and can therefore have different sizes. The use of fixed-distortion code books for discrete utterance classification was suggested by Rabiner [14]. Because each fixed-distortion code book is only as large as necessary to satisfy the distortion criterion, it follows that fixed-distortion code books might lead to the same classification performance as fixed-size code books but with fewer total codewords. This in turn would lead to smaller memory requirements and faster classification performance. Furthermore, fixed-distortion code books have the intuitive advantage that they are approximately equal in terms of a measure that should be connected closely to classification performance—average distortion in classifying the known training sequence. This measure may not, however, predict how well the code book set can discriminate among the different words in the recognition vocabulary. For both fixed-size and fixed-distortion code books, code book sizes are limited for convenience to powers of 2, i.e., $N_k = 2^{r_k}$, where r_k is the *rate* of $\boldsymbol{C}_k$—typical classification code books contain on the order of 32 codewords ($rk = 5$).

The second basic type of code book we used is the *unclustered code book*. These are generated simply by mak-

ing a codeword out of each frame in the training sequence. Training sequences for unclustered code books usually are much shorter than training sequences for clustered code books. Our motivation for considering unclustered code books was computational efficiency and convenience—generating them obviously is much easier than generating clustered code books. Although we did not expect them to perform as well as clustered code books, we thought that their performance might be sufficiently good for some applications.

In generating clustered code books, we used both d_{IS} and d_{GN} as the distortion measure in (2). We didn't use d_{GO} because it led to difficulties in computing cluster centroids—in the case of d_{IS} and d_{GN}, the centroid of a set of spectra can be computed simply by averaging the autocorrelations of the members (see [6], [8]). We refer to clustered code books generated with d_{IS} and d_{GN} respectively as IS and GN code books. In performing classification of unknown utterances, we used all three distortion measures (d_{IS}, d_{GN}, and d_{GO}) as the distortion measure in (3), except that d_{IS} was used only in the case of IS code books. (GN code books are generated by setting the gain to 1 before computing centroids, so it doesn't make sense to use the gain-sensitive measure d_{IS}.)

Spectrum shapes that result from analyzing nearly silent frames can be quite arbitrary. In order to avoid cluttering up code books with codewords that that would result from including such frames, we used an energy threshold during code book generation—frames with energy below the threshold were ignored. Similarly, another threshold was used to ignore the low-energy frames of an input utterance during classification.

C. Figures of Merit

We used two figures of merit in evaluating the experiments. The first is simply the number of classification errors made. The second attempts to quantify the extent to which the classifications are correct or incorrect. In particular, suppose that the input utterance is the mth word in the recognition vocabulary. For correct classification, D_m should be the smallest of the average distortions (3), i.e., $D_r = D_m$ (see (4)). Define

$$D^* = \min_{k \neq m} D_k \tag{11}$$

as the smallest average distortion of all code books except the correct one, and define

$$F = \frac{D^* - D_m}{D_m}. \tag{12}$$

If the classification is correct, $F > 0$; if the classification is incorrect, $F < 0$. For correct classifications, F is the fractional difference between the distortion of the correct code book, and the distortion of the next best choice—a large value of F means that the correct code book stands out clearly from the other choices. For each experiment, we computed the number of errors, the average value of F (F_{av}), and the standard deviation of F (F_σ).

D. Experiments

In this subsection we list the various experiments reported in the remainder of the paper. Section IV contains a brief summary of some previously reported experiments. The rest of the paper reports on experiments with the data base used in the test of commercial recognizers that we mentioned earlier [2]. The data base is described in Section V-A, and our experimental parameters are defined in Section V-B. The rest of Section V contains the results of speaker-dependent classification tests on all 16 speakers in the data base using fixed parameters that we selected after studying a single speaker. In particular, we performed the following experiments, which are listed according to the corresponding subsection of Section V:

C. Single-speaker (WMF) study of classification performance versus code book rate for fixed-size IS and GN code books and various classification distortion measures (Fig. 1);
Single-speaker (WMF) study of classification performance versus autoregressive model order for fixed-size code books (Fig. 2);
Single-speaker (WMF) study of classification performance versus average distortion for fixed-distortion IS and GN code books (Figs. 3–4);

D. Full-vocabulary classification performance for all 16 speakers using fixed parameters based on the WMF study (Tables I–III);
Digit-subset classification performance for all 16 speakers using fixed parameters based on the WMF study (Tables IV–VI).

Section VI contains the results of various speaker-dependent performance studies. The following experiments are listed according to the corresponding subsection of Section VI:

A. Classification performance versus code book rate for fixed-size IS and GN code books (Table VII);
Classification performance versus average distortion for fixed-distortion IS and GN code books (Figs. 5–6);

B. Classification performance versus fixed-size code book rate for classification by average distortion divided by span fraction (Table VIII);

C. Results for data downsampled to 8000 samples per second from 12 500 samples per second (Table IX);

D. Full-vocabulary results for classification with unclustered code books (Tables X and XII);
Digit-subset results for classification with unclustered code books (Table XIII);

E. Results for classification for power-normalized data (Table XIV).

Section VII contains results of the following multiple-speaker and speaker-independent experiments.

A. Results for classification of 4 speakers using a fixed-size code book set designed from utterances of all four speakers (Table XVI);

B. Single-speaker (RLD) study of speaker-independent classification performance versus fixed-size code book rate, versus size of training sequence, and versus method of estimating autoregressive model (autocorrelation or Burg technique) (Table XVII);
C. Full-vocabulary, speaker-independent classification performance for 8 male speakers using parameters based on the RLD study and two training sequence lengths (Table XVIII–XX);
Digit-subset classification performance for 8 male speakers using parameters based on the RLD study and two training sequence lengths (Tables XXI–XXIII).

IV. Summary of Previous Experiments

In this section we summarize some previous experiments in speaker-dependent classification; utterances from individual speakers were classified using code books designed from training sequences spoken by the same speaker. These experiments, conducted using a small data base, were reported in [10]. The experiments used a data base of the digits ZERO through NINE, each spoken eleven times by one male speaker—110 utterances in all. The beginning and end of each utterance were marked by hand. The speech was passed through an anti-aliasing filter, sampled at 6500 samples/s, and divided into frames of 128 samples (approximately 20 ms). Tenth order LPC analysis was performed on each frame by means of Levinson recursion on autocorrelations that were estimated with Hamming windowing and 90 percent preemphasis. Eight utterances were used as a training sequence in designing a fixed-size, rate-4 (size 16), IS code book for each word in the vocabulary. This left three utterances of each word outside of the training sequences. The thirty total utterances outside of the training sequences were classified using the rate-4 code books and d_{GO} for d in (3). No errors were made, with $F_{av} = 1.20$. The thirty utterances were classified again, this time using d_{IS} in (3). The results were similar: no errors were made, with $F_{av} = 1.05$.

We also performed limited studies with one- and four-utterance unclustered code books. Classification with d_{GO} was much better than with d_{IS}, and increasing the size of the unclustered code books improved performance. The unclustered code books, which were all larger than the clustered code books, generally performed worse than the clustered code books. For details, see [10].

V. Speaker-Dependent Performance on Large Data Base

All of our subsequent experiments were conducted using a much larger data base that was prepared by Texas Instruments, Inc. (TI), during a systematic test of existing discrete utterance recognition systems [2]. Some preliminary results were reported in [11].

Such a data base can be useful in both tuning and testing a recognition algorithm. In order to balance the conflict between tuning and unbiased testing, we chose the following procedure: we tuned the algorithm based on prior experience and on a preliminary study using one of the speakers in the TI data base. We then tested the results (without changing parameters) on the other speakers in the data base. The results of that study are reported in this section. Subsequently, we varied some of the parameters and performed studies on the entire data base in order to gain additional insights. The results of those studies are reported in subsequent sections.

A. TI Data Base

The TI data base [2] consists of twenty words: the digits ZERO through NINE and the ten control words YES, NO, ERASE, RUBOUT, REPEAT, GO, ENTER, HELP, STOP, and START. Eight male and eight female speakers each recorded twenty-six repetitions of each word in the vocabulary, for a total of 8320 utterances. The data was recorded on analog tape under tightly controlled conditions: the noise level was low, the speech level was restricted to a ± 3 dB range, the acoustic environment was unvarying and all errors in the input words were eliminated. After collection, the data was low-pass filtered and sampled at 12 500 samples per second. We received the data in digital form on magnetic tape. Each utterance, preceded and followed by short segments of ambient noise, was contained in a separate file.

We used automatic endpoint detection for both training-sequence and classification utterances in all of our experiments with the TI data base. Our endpoint-detection algorithm is based upon ideas described in [26], [27]. The algorithm first analyzes the background noise to determine its average magnitude A, and then uses the results to set three thresholds for the average adjusted magnitude—the average of $|S(t) - A|$ over 10 ms, where $S(t)$ is the speech waveform. The three thresholds—a start threshold, a high threshold, and an end threshold—are used to find significant "energy clumps" in the data. The end threshold is required to be lower than the start threshold. Briefly, a word is detected when the average adjusted magnitude satisfies the following sequence of criteria:

1) it rises above the start threshold;
2) it remains above the start threshold until it rises above the high threshold;
3) it drops below the end threshold; and
4) it doesn't exceed the start threshold for another 150 ms.

When all four criteria are satisfied, the endpoints are defined as the points at which (1) and (3) were satisfied. If (1)–(3) are satisfied, but (4) is not, the algorithm applies (3) and (4) again starting at the point where (4) failed. The purpose of this correction is to avoid being confused by short periods of low energy that occur in the middle of certain words (e.g., REPEAT).

Like the algorithm in [27], ours does not use zero-crossing information to try and distinguish between noise and sibilant speech. For this reason, the algorithm should also work well on lowpass-filtered speech such as exists in telephone channels or military communication channels.

B. Experimental Parameters

In this subsection we describe the various experimental parameters associated with code book generation and utterance classification. The parameters associated with code book generation are as follows:

a) code book type (clustered or unclustered);
b) number of utterances in the training sequence;
c) energy threshold $E_{\min}$, where E is computed by

$$E = \sum_{i=1}^{W} x_i^2$$

(Here, W is the analysis window width (see below), and x_i are the time-domain samples of the TI data after optional preemphasis and Hamming windowing.); and
d) LPC analysis parameters for determining inverse filter gain and sample coefficients of autoregressive models (see (6)).

For clustered code books, we also have

e) distortion measures used in clustering (d_{IS} or d_{GN});
f) method of determining code book size (fixed-rate or fixed-distortion); and
g) code book rate (for fixed-rate code books) or maximum average distortion (for fixed-distortion code books).

The parameters associated with utterance classification are as follows:

a) LPC analysis parameters for determining inverse filter gain and sample coefficients;
b) energy threshold; and
c) distortion measure (d_{IS}, d_{GO}, or d_{GN}).

The LPC analysis parameters for autoregressive modeling, relevant both to code book generation and to utterance classification, are

a) analysis method (autocorrelation or Burg);
b) number of points to shift between successive speech frames (N);
c) analysis window width (number of points within a speech frame that contribute to the analysis of the frame) (W);
d) data window (rectangular or Hamming);
e) preemphasis factor (0–98 percent); and
f) filter order (M).

For consistency, the LPC analysis parameters used in classifications were always chosen to match those used in generating the code books.

C. Parameter Selection

For the initial study, we selected clustered code books with the first 10 utterances of each word as the training sequences. (This choice for the training sequences is the same as that in [2].) The TI data has little background [illegible] after examining a few utterances. We selected the autocorrelation method of LPC analysis using Levinson recursion, along with $N = W = 250$ (20 ms), Hamming windowing, and 90 percent preemphasis.

Because we believed that classification performance would depend strongly on the remaining parameters, we conducted a preliminary study on a single speaker (WMF) to determine them. In particular, we investigated 1) code book size and method of determining size; 2) LPC filter order (M); and 3) distortion measures for clustering and classification.

We began by designing rate-2 through rate-6, fixed-size code book sets with both d_{IS} and d_{GN}. In each case, the first ten utterances of each word were used as training sequences. The remaining 320 (total) words were classified by all five code book sets using d_{IS}, d_{GO}, and d_{GN} with the IS code books and using d_{GO} and d_{GN} with the GN code books. The LPC filter order was fixed at $M = 16$. The results are shown in Fig. 1. Three trends are apparant:

1) independent of distortion measure types, the error rate tends to decrease with increasing code book rate;
2) classification using d_{GO} yields the best classification performance; and
3) increasing the code book rate decreases the performance differences resulting from the various distortion measures.

Next, we studied the effect of analysis filter order (M). We generated fixed-size, rate-5, IS and GN code book sets for $M = 8$, 10, 12, 14, 18, and 20. Only d_{GO} was used as a classification distortion measure. The results are shown in Fig. 2. For a particular value of M the results for IS code books were usually better than those for GN code books, but both IS and GN code books exhibited the same general trend: the error rates initially decreased smoothly with increasing M and became constant for large M.

Next, we evaluated the choice between fixed-size and fixed-distortion code books. We designed fixed-distortion IS and GN code book sets with five different distortion thresholds and with parameter settings otherwise the same as those for the fixed-size study that was summarized in Fig. 1, except that only d_{GO} was used as a classification distortion measure. The performance of the IS code book sets is plotted in Fig. 3 as a function of the actual average distortion of the code book set. (The actual average is different from the design threshold because the code book sizes are limited to powers of 2.) Also plotted in Fig. 3 is the performance of the fixed-size, IS code book sets from Fig. 1 for the case of classification with d_{GO}. For these fixed-size code book sets, the performance is plotted as a function of the average distortion with which the code books in the set encoded their training sequences. For both the fixed-distortion and fixed-size code book sets, the average code book size appears in parentheses next to the plotted point. Note that all code books in a fixed-size code book set have the size indicated. Analogous results for GN code book sets are shown in Fig. 4. Comparing fixed-size

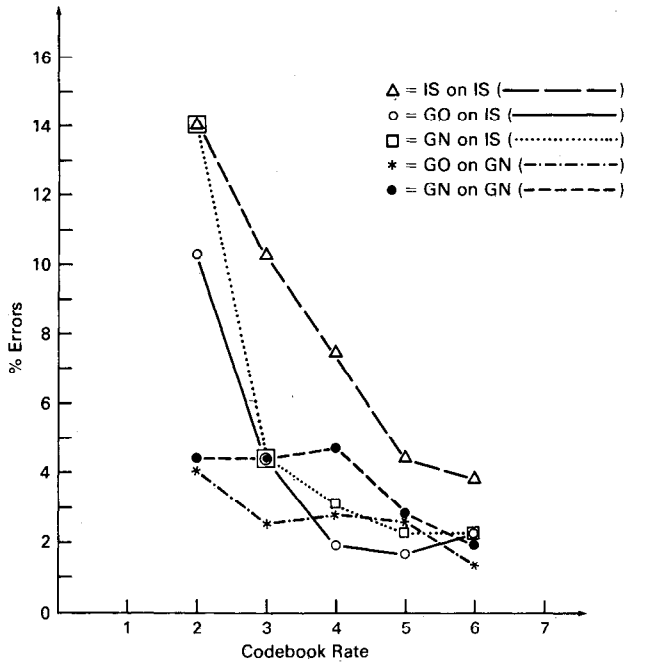

Fig. 1. Single-speaker study of error rate versus code book rate. LPC filter order is 16. "IS on IS" means classifications were performed using d_{IS} with IS code books; "GO on IS" means classifications were performed using d_{GO} with IS code books; "GN on IS" means classifications were performed using d_{GN} with IS code books; "GO on GN" means classifications were performed using d_{GO} with GN code books; and "GN on GN" means classifications were performed using d_{GN} with GN code books.

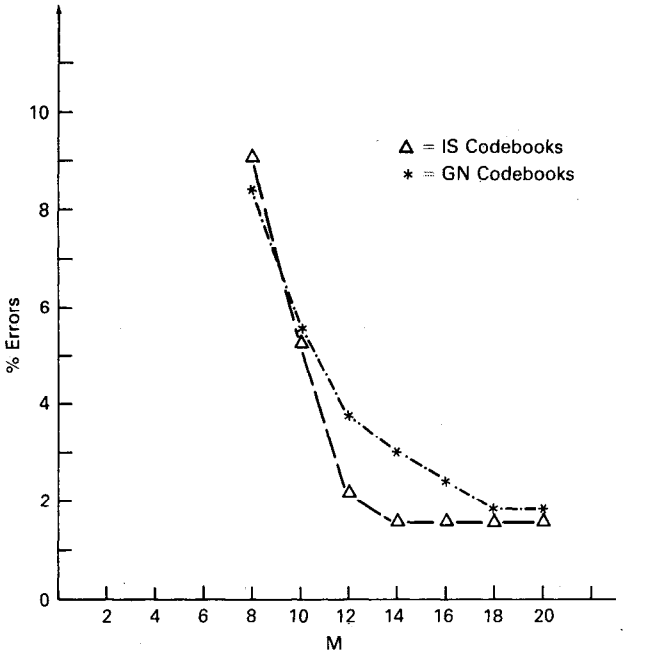

Fig. 2. Single-speaker study of error rate versus LPC filter order (M). Code book rate is 5. Classification distortion measure is d_{GO}.

approximately the same average code book size, it is apparent that the fixed-size code books performed better. We therefore selected fixed-size code books for the full data base experiment reported in the next subsection. Since the results in Figs. 3–4 are somewhat counterintuitive, however, we explored the issue again in studies on the full data base reported in Section VI.

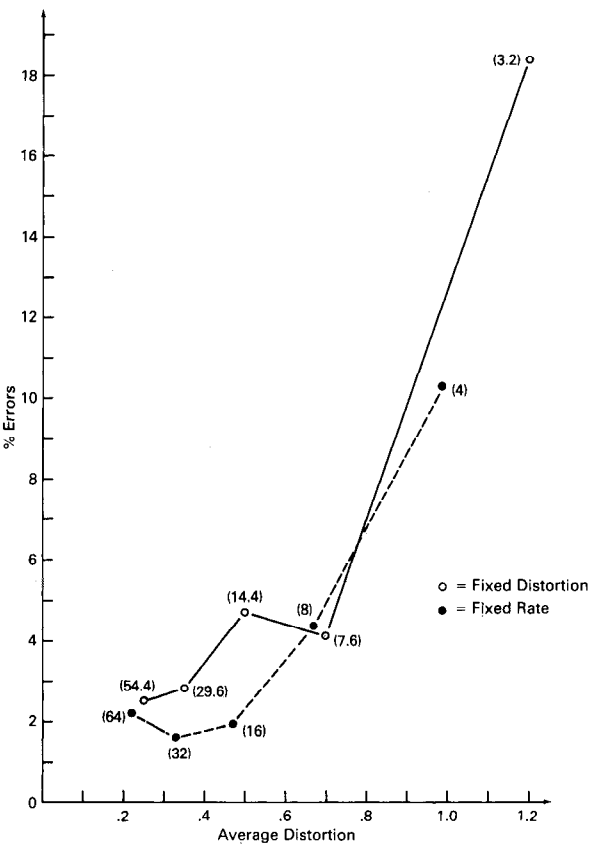

Fig. 3. Single-speaker study of error rate versus average code book distortion using IS code books. Classification distortion measure is d_{GO}. LPC filter order is 16. For each code book set, the average number of codewords per code book is shown in parenthesis.

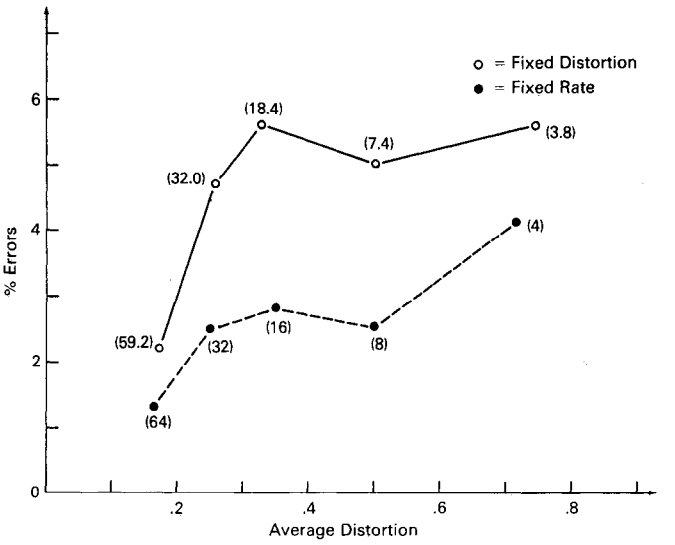

Fig. 4. Single-speaker study of error rate versus average code book distortion using GN code books. Classification distortion measure is d_{GO}. LPC filter order is 16. For each code book set, the average number of codewords per code book is shown in parenthesis.

D. Results for the Full Data Base

For the initial study on the entire data base, we used the same parameter settings that we fixed for the preliminary study on WMF: clustered code books with 10-utterance

TABLE I
INITIAL RESULTS FOR FULL TI DATA BASE

Speaker	No. Class	IS Code Books				GN Code Books			
		Errors	% Right	F_{av}	F_σ	Errors	% Right	F_{av}	F_σ
WMF	320	5	98.4	0.499	0.328	8	97.5	0.475	0.302
RLD	320	1	99.7	0.653	0.466	2	99.4	0.664	0.437
RGL	320	0	100.0	0.909	0.431	0	100.0	0.894	0.423
MSW	320	6	98.1	0.618	0.372	0	100.0	0.592	0.337
GRD	320	4	98.8	0.592	0.384	4	98.8	0.590	0.378
TBS	320	13	95.9	0.606	0.388	7	97.8	0.591	0.346
KAB	320	4	98.8	0.615	0.369	8	97.5	0.618	0.392
REH	320	2	99.4	0.825	0.503	1	99.7	0.852	0.511
CJP	320	4	98.4	0.629	0.323	1	99.7	0.648	0.325
DFG	320	5	98.4	0.612	0.338	1	99.7	0.621	0.329
GNL	320	2	99.4	0.883	0.554	2	99.4	0.897	0.542
JWS	320	2	99.4	0.941	0.570	1	99.7	0.947	0.568
HNJ	320	4	98.8	0.830	0.494	3	99.1	0.836	0.469
SAS	320	3	99.1	0.798	0.423	2	99.4	0.804	0.428
SJN	320	0	100.0	0.909	0.502	0	100.0	0.931	0.501
ALK	320	1	99.7	0.723	0.409	0	100.0	0.693	0.410
all	5120	56	98.9	0.728	0.456	40	99.2	0.728	0.450

training sequences, $E_{min} = 250$, and LPC analyses by means of Levinson recursion on autocorrelations estimated with Hamming windowing and 90 percent preemphasis. Based on the results of the WMF study, we chose fixed-size, rate-5 (size 32), IS and GN code books, LPC order $M = 16$, and classification distortion measure d_{GO}.

Shown in Table I are the results for all 16 speakers for both the IS and GN code books. The first 8 speakers are male, the rest are female. The performance for both code book types was approximately the same—error rates of about 1 percent. Although we are testing an algorithm as opposed to a commercial device, it is worth noting that these results are better than six of the seven commercial devices tested with the same data base [2]. The results show that much more can be done without time-sequence information than is commonly assumed.

The results in Table I exhibit the so-called "goat-sheep" phenomenon [2], in which a large fraction of the errors occur within a small segment, the "goats," of the population. For the GN code books, over half of the errors occurred for just three speakers: KAB, TBS, and WMF. WMF and TBS were also hard speakers for the IS code books. Contrary to what is usually found [2], our results were better for female speakers than they were for male speakers, and we have no explanation for this trend.

In Tables II and III, we present summaries of the specific errors in the form of confusion matrices for the total IS and GN code book results. Each row comprises the results for one word in the recognition vocabulary; the columns correspond to the different classification decisions. Each row contains the results of classifying all utterances of one word in the vocabulary.

Some of the error classes shown in the confusion matrices are easily understandable. For example, the NO ↔ ONE confusions are not surprising—if a recording of NO is played backwards, it sounds like ONE and vice versa. It follows that the two words have many similar spectra and could be confused by a method that ignores time sequence information. This is an example of a general class of potential confusions. The GO ↔ NO confusion is another example of this general class. Because there is a time-sequence similarity as well, it is a confusion that occurs with many other methods of word recognition. The unilateral confusions like SIX → YES are probably examples of the same phenomenon—more of the spectra in SIX are similar to spectra in YES than vice versa. As we mentioned in Section III, one possible way to reduce confusions between words that comprise similar spectra is to classify by minimizing average distortion divided by span fraction. This possibility was the basis for one of the studies reported in Section VI.

Since many word-recognition applications involve only the digits ZERO through NINE, we also obtained results for this restricted case. The results for each speaker are shown in Table IV, with summary confusion matrices in Tables V and VI. As one would expect, the digit results are much better than those for the full vocabulary. For the digits, there was an extreme example of a "goat"—nine of the ten errors with IS code books and five of the eight errors with GN code books occurred for one male speaker (TBS).

VI. SPEAKER-DEPENDENT STUDIES

In order to gain additional insights into the method in general as well as into the effects of the various parameters, we performed a variety of additional studies. In particular, we studied performance as a function of code book size, IS versus GN code books, fixed-size versus fixed-distortion code books, average distortion classification versus average distortion divided by span fraction, clustered versus unclustered code books, performance on 4 kHz. bandwidth data, and performance on power-normalized data.

A. Clustered Code Book Type and Classification Measure

We repeated the experiment described in Section V-D using fixed-size, IS and GN code book sets with rates 2, 3,

TABLE II
FULL DATA BASE CONFUSION MATRIX FOR IS CODE BOOKS

	0	1	2	3	4	5	6	7	8	9	ENTER	ERASE	GO	HELP	NO	RUBOUT	REPEAT	STOP	START	YES
0	256	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
1	·	253	·	·	·	1	·	·	·	·	·	·	·	·	2	·	·	·	·	·
2	·	·	255	·	·	·	·	·	·	·	1	·	·	·	·	·	·	·	·	·
3	·	·	·	253	·	·	·	·	·	·	·	·	·	·	·	·	3	·	·	·
4	·	·	·	·	256	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
5	·	·	·	·	·	251	·	1	·	·	·	·	·	1	·	3	·	·	·	·
6	·	·	·	·	·	·	248	3	·	·	·	2	·	·	·	·	·	·	·	3
7	·	·	·	·	·	·	·	256	·	·	·	·	·	·	·	·	·	·	·	·
8	·	·	·	·	·	·	·	·	252	·	·	·	·	·	·	·	4	·	·	·
9	·	·	2	·	·	·	2	·	1	251	·	·	·	·	·	·	·	·	·	·
ENTER	·	·	·	·	·	·	·	·	·	·	256	·	·	·	·	·	·	·	·	·
ERASE	·	·	·	·	·	·	·	·	·	·	·	256	·	·	·	·	·	·	·	·
GO	·	·	·	·	·	·	·	·	·	·	·	·	242	·	13	1	·	·	·	·
HELP	·	·	·	·	·	·	1	·	·	·	·	·	·	255	·	·	·	·	·	·
NO	·	·	4	·	·	·	·	·	·	·	·	·	3	·	249	·	·	·	·	·
RUBOUT	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	256	·	·	·	·
REPEAT	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	256	·	·	·
STOP	·	1	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	252	3	·
START	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	256	·
YES	·	·	·	·	·	·	·	·	·	·	·	·	1	·	·	·	·	·	·	255

TABLE III
FULL DATA BASE CONFUSION MATRIX FOR GN CODE BOOKS

	0	1	2	3	4	5	6	7	8	9	ENTER	ERASE	GO	HELP	NO	RUBOUT	REPEAT	STOP	START	YES
0	256	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
1	·	254	·	·	·	·	·	·	·	·	·	·	·	·	2	·	·	·	·	·
2	·	·	255	1	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
3	·	·	·	254	·	·	·	·	·	·	·	·	·	·	·	·	2	·	·	·
4	·	·	·	·	256	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
5	·	·	·	·	·	247	·	·	·	·	·	·	·	5	·	4	·	·	·	·
6	·	·	·	·	·	·	252	1	·	·	·	1	·	·	·	·	·	·	·	2
7	·	·	·	·	·	·	·	256	·	·	·	·	·	·	·	·	·	·	·	·
8	·	·	·	·	·	·	·	·	255	·	·	·	·	·	·	·	·	·	·	1
9	·	·	2	·	·	·	1	·	·	253	·	·	·	·	·	·	·	·	·	·
ENTER	·	·	·	·	·	·	·	·	·	·	256	·	·	·	·	·	·	·	·	·
ERASE	·	·	·	·	·	·	·	·	·	·	·	256	·	·	·	·	·	·	·	·
GO	1	·	·	·	·	·	·	·	·	·	·	·	247	1	7	·	·	·	·	·
HELP	·	·	·	·	·	·	·	·	·	·	·	·	·	256	·	·	·	·	·	·
NO	·	5	·	·	·	·	·	·	·	·	·	·	·	2	249	·	·	·	·	·
RUBOUT	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	256	·	·	·	·
REPEAT	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	256	·	·	·
STOP	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	255	1	·
START	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	256	·
YES	·	·	·	·	·	·	·	·	·	·	·	1	·	·	·	·	·	·	·	255

TABLE IV
INITIAL RESULTS FOR DIGIT SUBSET

		IS Code Books				GN Code Books			
Speaker	No. Class	Errors	% Right	F_{av}	F_σ	Errors	% Right	F_{av}	F_σ
WMF	160	1	99.4	0.714	0.371	2	98.8	0.670	0.364
RLD	160	0	100.0	1.043	0.543	0	100.0	1.059	0.550
RGL	160	0	100.0	1.144	0.507	0	100.0	1.117	0.503
MSW	160	0	100.0	0.923	0.458	0	100.0	0.901	0.432
GRD	160	0	100.0	0.915	0.459	0	100.0	0.943	0.466
TBS	160	9	94.4	0.850	0.510	5	96.9	0.810	0.444
KAB	160	0	100.0	0.970	0.511	1	99.4	0.970	0.520
REH	160	0	100.0	1.255	0.686	0	100.0	1.239	0.710
CJP	160	0	100.0	0.983	0.462	0	100.0	0.996	0.459
DFG	160	0	100.0	0.923	0.378	0	100.0	0.934	0.412
GNL	160	0	100.0	1.428	0.702	0	100.0	1.424	0.677
JWS	160	0	100.0	1.540	0.845	0	100.0	1.512	0.816
HNJ	160	0	100.0	1.215	0.614	0	100.0	1.188	0.576
SAS	160	0	100.0	1.138	0.507	0	100.0	1.146	0.549
SJN	160	0	100.0	1.441	0.698	0	100.0	1.412	0.690
ALK	160	0	100.0	1.200	0.607	0	100.0	1.124	0.542
all	2560	10	99.6	1.105	0.611	8	99.7	1.090	0.600

TABLE V
DIGITS-ONLY CONFUSION MATRIX FOR IS CODE BOOKS

	0	1	2	3	4	5	6	7	8	9
0	256	·	·	·	·	·	·	·	·	·
1	·	255	·	·	·	1	·	·	·	·
2	·	·	256	·	·	·	·	·	·	·
3	·	·	·	256	·	·	·	·	·	·
4	·	·	·	·	256	·	·	·	·	·
5	·	·	·	·	·	255	·	1	·	·
6	·	·	·	·	·	·	253	3	·	·
7	·	·	·	·	·	·	·	256	·	·
8	·	·	·	·	·	·	·	·	256	·
9	·	·	2	·	·	·	2	·	1	251

TABLE VI
DIGITS-ONLY CONFUSION MATRIX FOR GN CODE BOOKS

	0	1	2	3	4	5	6	7	8	9
0	256	·	·	·	·	·	·	·	·	·
1	·	256	·	·	·	·	·	·	·	·
2	·	·	255	1	·	·	·	·	·	·
3	·	·	·	256	·	·	·	·	·	·
4	·	·	·	·	256	·	·	·	·	·
5	·	·	·	·	·	253	·	2	·	1
6	·	·	·	·	·	·	255	1	·	·
7	·	·	·	·	·	·	·	256	·	·
8	·	·	·	·	·	·	·	·	256	·
9	·	2	·	·	·	1	·	·	·	253

4, and 6. The overall results and those from the previous rate-5 experiment are shown in Table VII. The GN code books generally performed better, but the difference was substantial only for code book rates 2 and 3. The computational complexity of generating IS and GN code books is essentially the same, but the GN code books require no gain term so they require slightly less storage. Perhaps the most surprising result in Table VII is the remarkably good performance of the rate-2 code book sets. With only four codewords per code book, the GN code books had an error rate of only 2.3 percent.

Next, we repeated the experiment using fixed-distortion IS and GN code book sets that were designed to five different average distortion thresholds. The results and those for the fixed-size code book sets (Table VII) are plotted in Figs. 5–6—overall performance is plotted as a function of the average distortion with which the code books in the set encoded their training sequences. Each point in Figs. 5–6 is based on the classification of 5120 utterances outside the training sequences. The results confirm the conclusion of the WMF study (Figs. 3–4)—fixed-size code books perform better with fewer total codewords. For example, for almost every fixed-distortion point in Fig. 6, one can find a fixed-size point with higher average distortion but with fewer average codewords per code book and lower error rate. The higher average distortion of the fixed-size code books is reasonable since they have fewer average codewords per code book than the fixed-distortion code books. The lower error rate of the fixed-size code books is, however, puzzling.

An examination of the errors showed that the fixed-distortion code books performed slightly better on some mul-

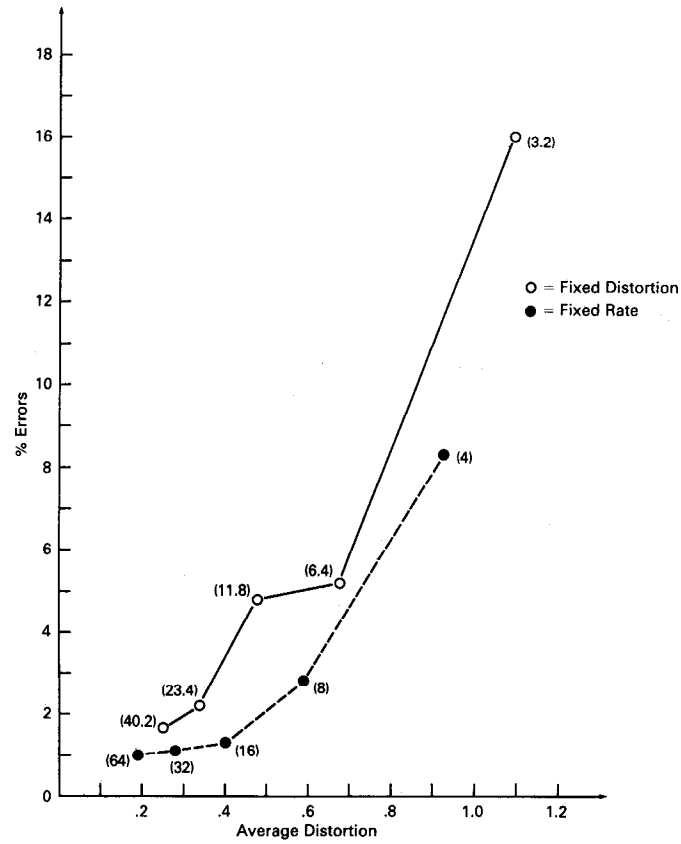

Fig. 5. Full data base study of error rate versus average code book distortion using IS code books. Classification distortion measure is d_{GO}. LPC filter order is 16. For each code book set, the average number of codewords per code book is shown in parenthesis.

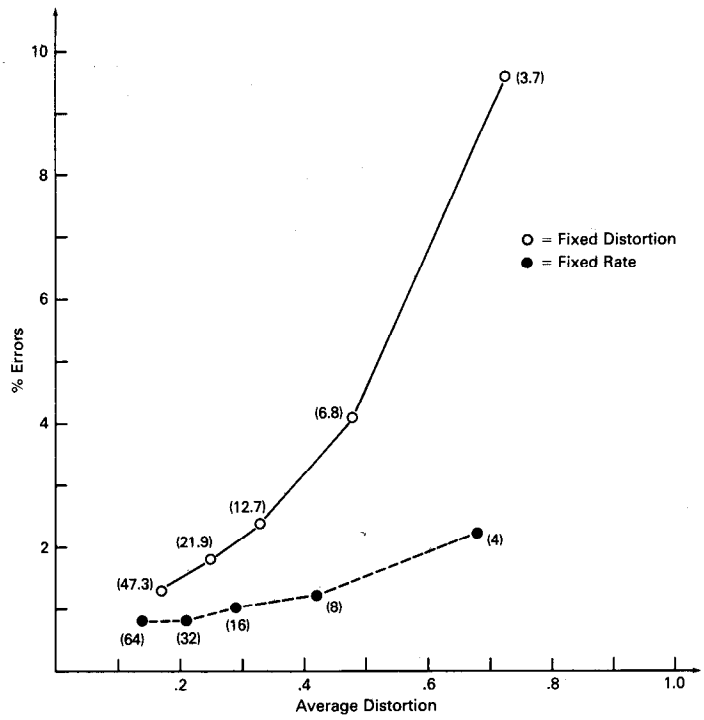

Fig. 6. Full data base study of error rate versus average code book distortion using GN code books. Classification distortion measure is d_{GO}. LPC filter order is 16. For each code book set, the average number of codewords per code book is shown in parenthesis.

TABLE VII
RESULTS OF RATE STUDY FOR FIXED-SIZE CODE BOOKS

Code Book Rate	No. Class	IS Code Books				GN Code Books			
		Errors	% Right	F_{av}	F_σ	Errors	% Right	F_{av}	F_σ
2	5120	423	91.7	0.342	0.301	116	97.7	0.447	0.294
3	5120	144	97.2	0.519	0.373	62	98.8	0.597	0.373
4	5120	69	98.7	0.647	0.416	52	99.0	0.684	0.422
5	5120	56	98.9	0.728	0.456	40	99.2	0.728	0.450
6	5120	52	99.0	0.765	0.472	41	99.2	0.745	0.455

TABLE VIII
RESULTS OF SPAN FRACTION STUDY

Code Book Rate	No. Class	IS Code Books				GN Code Books			
		Errors	% Right	F_{av}	F_σ	Errors	% Right	F_{av}	F_σ
2	5120	460	91.0	0.382	0.340	138	97.3	0.504	0.348
3	5120	224	95.6	0.709	0.530	123	97.6	0.810	0.537
4	5120	99	98.1	1.018	0.706	78	98.5	1.054	0.684
5	5120	82	98.4	1.155	0.780	73	98.6	1.124	0.756
6	5120	76	98.5	1.161	0.799	64	98.8	1.100	0.749

tisyllabic words, but considerably worse on some of the monosyllabic words. The confusions that the fixed-distortion code books made on monosyllabic words were generally a superset of those made by the fixed-size code books. The additional confusions made by the fixed-distortion code books were usually into multisyllabic words that corresponded to code books containing about twice as many codewords as those of the monosyllabic words, the most frequent examples being FIVE → RUBOUT, SIX → ERASE, THREE → REPEAT, and EIGHT → REPEAT. It seems that the spectra in these larger code books were sufficiently varied so that a low average distortion resulted from encoding some of the shorter words in the recognition vocabulary. As pointed out by an anonymous referee, this explanation suggests that classification by minimizing average distortion divided by span fraction might work better for fixed-distortion code books than for fixed-size code books.

Our results concerning fixed-size versus fixed-distortion code books points out that the key to high recognition accuracy is how well the code books discriminate among the vocabulary words, not simply how well the training sequence is characterized. Apparently, fixed-size code books discriminate better than fixed-distortion code books.

B. Average Distortion Versus Span Fraction

We repeated the fixed-size code book set experiments (Table VII) with identical parameters except that classification was performed by minimizing average d_{GO} divided by span fraction instead of just average d_{GO}. The results, shown in Table VIII, are always worse than the corresponding results in Table VII.

In response to the referee's comment mentioned in the previous subsection, we repeated the fixed-distortion code book experiments described in that subsection, except that classification was performed by minimizing average distortion divided by span fraction. Indeed, some improvement resulted. In terms of the results plotted in Figs. 5–6, the performance improved slightly for the fixed-distortion code books designed to distortion thresholds greater than about 0.4—below this point the performance was worse. In no case, however, did performance improve to the level of the equivalent fixed-rate code book.

C. Performance on 4 kHz Bandwidth Data

The excellent results shown in Tables I–VII were for high-fidelity data sampled at 12 500 samples/s. This represents a larger bandwidth than available over most commercial and military telephone and radio telephone networks. Our original experiments, reported in [10] and summarized in Section IV, were performed with data that was sampled at 6500 samples/s. The performance of d_{GO} classification using fixed-size, rate-4 code books was good, and it implied that our technique could be useful with telephone-bandwidth speech, but the data base was extremely small (a total of 110 utterances). To obtain more data on the classification of telephone-bandwidth speech, we performed some classification experiments on the TI data base after down sampling the data to 4000 Hz. The sampling rate conversion was carried out by [28]:

1) padding the signal with zeros to create a signal sampled at 200 000 samples/s;
2) lowpass filtering the signal at 3900 Hz; and
3) down sampling the signal to 8000 samples/s.

The interpolation/decimation filter consisted of a fourth-order elliptical filter cascaded with a third-order Chebyshev filter. After converting the sampling rate, we ran the endpoint detection algorithm described in Section V-A.

The 4000 Hz bandwidth and the following LPC analysis conditions were chosen for compatibility with the Navy's 2.4-kbs LPC-10 [29]: $N = 180$, $W = 128$, $M = 10$, and Hamming windowing. We continued to use Levinson recursion, although LPC-10 uses the covariance method. As

TABLE IX
RESULTS FOR 4 kHz BANDWIDTH DATA

Speaker	No. Class	IS Code Books				GN Code Books			
		Errors	% Right	F_{av}	F_σ	Errors	% Right	F_{av}	F_σ
WMF	320	7	97.8	0.738	0.491	5	98.4	0.720	0.464
GRD	320	5	98.4	0.659	0.481	6	98.1	0.687	0.499
CJP	320	2	99.4	0.758	0.436	2	99.4	0.773	0.429
JWS	320	5	98.4	1.114	0.745	2	99.4	1.167	0.743
all	1280	19	98.5	0.817	0.579	15	98.8	0.837	0.581

TABLE X
RESULTS FOR 10-UTTERANCE UNCLUSTERED CODE BOOKS

Speaker	No. Class	Avg. Size	d_{IS} Classification				d_{GO} Classification			
			Errors	% Right	F_{av}	F_σ	Errors	% Right	F_{av}	F_σ
WMF	320	225.8	21	93.4	0.532	0.389	3	99.1	0.518	0.312
TBS	320	248.3	18	94.4	0.585	0.464	7	97.8	0.618	0.336
RLD	320	225.0	6	98.1	0.658	0.457	1	99.7	0.663	0.449
CJP	320	321.0	1	99.7	0.812	0.388	2	99.4	0.669	0.336
all	1280	255.0	46	96.4	0.647	0.439	13	99.0	0.617	0.367

TABLE XI
RESULTS FOR FIXED-SIZE, RATE-5, CLUSTERED CODE BOOKS (FROM TABLE I)

Speaker	No. Class	IS Code Books				GN Code Books			
		Errors	% Right	F_{av}	F_σ	Errors	% Right	F_{av}	F_σ
WMF	320	5	98.4	0.499	0.328	8	97.5	0.475	0.302
TBS	320	13	95.9	0.606	0.388	7	97.8	0.591	0.346
RLD	320	1	99.7	0.653	0.466	2	99.4	0.664	0.437
CJP	320	4	98.8	0.629	0.323	1	99.7	0.648	0.325
all	1280	23	98.2	0.597	0.385	18	98.6	0.595	0.364

before, we generated fixed-size, rate-5 clustered IS and GN code books using 10-utterance training sequences, and an energy threshold of $E_{min} = 250$. We tested two male and two female speakers using average d_{GO} as a classification criterion. The results, which should be compared with rows in Table I that correspond to the same speakers, are shown in Table IX. For the 1280 classification trials, the error rates for the narrow-bandwidth tests are approximately equal to those of the full bandwidth tests. Although there was some overlap, different errors tended to be made in the tests at different bandwidths.

Based upon these results, we believe that our approach is capable of achieving a high level of performance on telephone or military-channel bandwidth data.

D. Results for Unclustered Code Books

Unclustered code books are generated simply by making a codeword out of every frame in the training sequence. Unclustered code books therefore can be viewed as a limiting case of clustered code books, namely when one requires that the resulting code book encode the training sequence with zero distortion. Since the clustering procedure attempts to find codewords that are representative of a training sequence, this suggests that the effectiveness of clustering can be evaluated by comparing the performance of clustered and unclustered code books designed from the same training sequence.

Accordingly, for four speakers we generated unclustered code books from the 10-utterance training sequences that were the basis of the fixed-size, rate-5 clustered code book experiments discussed in Section V and summarized in Table I. All other parameters were the same, except that we classified the utterances outside of the training sequences using both d_{IS} and d_{GO} instead of just d_{GO}. The results are shown in Table X. For ease of comparison, the appropriate four lines from Table I are collected in Table XI. Comparing Tables X and XI shows that classification using d_{GO} worked better with the unclustered code books than either the clustered IS or GN code books. The differences were small, however, which attests to the effectiveness of the clustering procedure—the clustered code books were about one eighth the size of the unclustered code books.

If one views the use of clustered code books as a method of condensing unclustered code books, the foregoing results suggest the potential of using other methods for condensing the training sequences, as has been done in some related work by Miclet and Nehame [30]. Note that unclustered code books are used as an initial guess in the k-means clustering technique [31]. One general class of methods for condensing the training sequence involves picking a subset that satisfies an optimality criterion. Possible methods include that of Hart [32]. Another class of methods involves picking an arbitrary subset, for example, by using fewer utterances in unclustered code book train-

TABLE XII
RESULTS FOR 1-UTTERANCE UNCLUSTERED CODE BOOKS

Speaker	No. Class	Avg. Size	d_{IS} Classification				d_{GO} Classification			
			Errors	% Right	F_{av}	F_σ	Errors	% Right	F_{av}	F_σ
WMF	320	21.9	132	58.8	0.100	0.529	31	90.3	0.339	0.280
TBS	320	23.9	142	55.6	0.081	0.500	41	87.2	0.481	0.462
RLD	320	24.4	83	74.1	0.235	0.397	30	90.6	0.366	0.326
CJP	320	30.4	41	87.2	0.439	0.415	14	95.6	0.408	0.264
all	1280	25.2	398	68.9	0.214	0.485	116	90.9	0.354	0.292

TABLE XIII
RESULTS FOR CLASSIFICATION OF DIGIT SUBSET WITH 1-UTTERANCE UNCLUSTERED CODE BOOKS

Speaker	No. Class	Avg. Size	d_{IS} Classification				d_{GO} Classification			
			Errors	% Right	F_{av}	F_σ	Errors	% Right	F_{av}	F_σ
WMF	160	22.4	77	51.9	0.303	0.975	14	91.3	0.462	0.316
TBS	160	23.3	64	60.0	0.276	0.768	11	93.1	0.496	0.338
RLD	160	24.4	30	81.3	0.494	0.535	3	98.1	0.677	0.398
CJP	160	30.9	4	97.5	0.602	0.362	0	100.0	0.657	0.278
all	640	25.3	175	72.7	0.419	0.713	28	95.6	0.573	0.348

TABLE XIV
RESULTS OF d_{IS} CLASSIFICATION USING FIXED-SIZE, RATE-5, CLUSTERED IS CODE BOOKS FOR POWER-NORMALIZED DATA

Speaker	No. Class	Normalized Data				Unnormalized Data			
		Errors	% Right	F_{av}	F_σ	Errors	% Right	F_{av}	F_σ
WMF	320	9	97.2	0.494	0.293	14	95.6	0.518	0.356
TBS	320	21	93.4	0.587	0.459	20	93.8	0.575	0.459
RLD	320	5	98.4	0.669	0.461	6	98.1	0.679	0.479
KAB	320	8	97.5	0.699	0.452	8	97.5	0.640	0.375
all	1280	43	96.6	0.612	0.430	48	96.3	0.603	0.425

ing sequences. The "random quantizers" in [33] are unclustered code books comprising a random selection of frames from the training sequence. Some limited results with 1-utterance unclustered code books in our original experiments suggested that surprisingly good performance could be obtained [10]. To obtain more data, we repeated the experiments reported in Table X, except that we used only one utterance from the training sequences for the unclustered code books. In each of the four experiments, the same 320 utterances as before were classified. The results are shown in Table XII. The results for classification with d_{GO} show that about 90 percent accuracy can be expected using single-utterance, unclustered code books. For the digit subset, accuracy increases to about 95 percent (Table XIII). Since unclustered code books are so easy to generate, these results are quite good. They suggest that the method could be used for an easy-to-program, easy-to-train, "poor-man's" discrete utterance speech recognizer.

E. Power-Normalized Itakura–Saito Classification

Classification using d_{IS} in (3) can only be used for code books that include gain terms, i.e., for IS clustered code books and unclustered code books. Results from Fig. 1, Table X, and Table XII show consistently that classification with d_{IS} in these cases is always inferior to classification with d_{GO}. These results are somewhat disturbing, since d_{IS} is a special case of asymptotic cross entropy [9], [7], [16], and since classification with d_{IS} is optimal in a well-defined information-theoretic sense [8].

The problem appears to be that small code books are not capable of reflecting the large gain variations that occur, a conjecture supported by the large difference in performance for d_{IS} classification between 1- and 10-utterance unclustered code books (see Tables X and XII). It follows that decreasing the gain variations in the training and classification data might improve the performance of d_{IS}. In an attempt to remove large gain variations for similar spectra, but still to allow different spectra to have characteristic gain terms, we normalized each utterance so that they all had the same average power. This power-normalized data was then used to build code books and used as classification data. We didn't expect dramatic differences since TI restricted the level of the speech data to within a ± 3 dB range when they recorded it.

We generated fixed-size, rate-5 clustered IS code books for four male speakers using the power-normalized utterances and parameter settings otherwise equivalent to the experiments reported in Table I. For each speaker, we classified the 320 utterances outside of the training sequence using d_{IS}. We then repeated the four experiments, except that we used the original, unnormalized data. The results are shown in Table XIV. For ease in comparing with the previous results for classification with d_{GO}, the

TABLE XV
RESULTS OF d_{GO} CLASSIFICATION USING FIXED-SIZE, RATE-5, CODE BOOKS (FROM TABLE I)

Speaker	No. Class	IS Code Books				GN Code Books			
		Errors	% Right	F_{av}	F_σ	Errors	% Right	F_{av}	F_σ
WMF	320	5	98.4	0.499	0.328	8	97.5	0.475	0.302
TBS	320	13	95.9	0.606	0.388	7	97.8	0.591	0.346
RLD	320	1	99.7	0.653	0.466	2	99.4	0.664	0.437
KAB	320	4	98.8	0.615	0.369	8	97.5	0.618	0.392
all	1280	23	98.2	0.593	0.395	25	98.1	0.587	0.379

TABLE XVI
RESULTS OF MULTIPLE SPEAKER EXPERIMENT

Speaker	No. Class	IS Code Books				GN Code Books			
		Errors	% Right	F_{av}	F_σ	Errors	% Right	F_{av}	F_σ
WMF	320	18	94.4	0.302	0.210	17	94.7	0.303	0.214
TBS	320	15	95.3	0.318	0.244	15	95.3	0.317	0.245
REH	320	8	97.5	0.478	0.332	7	97.8	0.509	0.341
GRD	320	22	93.1	0.365	0.283	19	94.1	0.353	0.257
all	1280	63	95.1	0.366	0.280	58	95.5	0.317	0.245

appropriate four lines from Table I are collected in Table XV. The results for d_{IS} classification with normalized data are still significantly worse than the results for d_{GO} classification with unnormalized data. The results for d_{IS} classification with normalized data may be slightly better than those for unnormalized data, but clearly there is no significant improvement.

VII. MULTIPLE-SPEAKER AND SPEAKER-INDEPENDENT EXPERIMENTS

We use the terms *multiple-speaker* and *speaker-independent* recognition when a set of speakers contribute to the training sequence of a code book. Multiple-speaker recognition is the case when the resulting code book is used for classifying utterances from these same speakers. Speaker-independent recognition is the case when the code book is used for classifying utterances from a speaker who is not in the training set.

In this section we present the results of one multiple-speaker classification test, several preliminary speaker-independent parameter studies, and a male speaker-independent experiment. The data and the LPC analysis conditions in these experiments were the same as in the speaker-dependent experiment discussed in Section V (Table I). The number of speakers in the training sequence, the number of utterances per speaker in the training sequence, and the code book rate were varied in the preliminary experiments.

A. Multiple-Speaker Test

To gain an appreciation for the degradation introduced by interspeaker variations in the training sequence, a four-speaker training sequence was used to generate IS and GN code books. In particular, fixed-size, rate-5 clustered code books were designed from two utterances each from REH, GRD, WMF, and TBS (all males). Additional utterances from the same four speakers were then classified using d_{GO}. The results are shown in Table XVI. As expected, classification accuracy is less than that of the speaker-dependent case (Table I), but the degradation is surprisingly small.

B. Speaker-Independent Parameter Studies

Using the same four speakers as in the multiple-speaker experiment, we generated four additional code book sets: fixed-size, rate-5 and rate-6 clustered IS and GN code books using six utterances per speaker in the training sequences. We then classified all 520 utterances from RLD, a male speaker not included in the training set, using d_{GO}. The classification accuracies for all six code book sets are shown in the first row of Table XVII. The results do not strongly favor any of the code book sets.

The second row in Table XVII was motivated by results in [34]. There, the Burg technique for estimating LPC parameters was used in speech coding by vector quantization. The Burg technique led to code books that encoded the training sequence with lower average distortion than the code books designed from LPC parameters estimated by the autocorrelation method. To see if the Burg technique would lead to better classification code books, we estimated LPC parameters and equivalent autocorrelations with the Burg technique, using 90 percent preemphasis, order $M = 16$, and frame size $N = 250$. We then generated new multiple-speaker code books using the results together with the usual energy threshold of $E_{min} = 250$, and we repeated the classification of RLD except that we again used Burg estimation for the autocorrelations. The results were about the same as those for the autocorrelation method.

For additional information, we compared the spectra in the code books designed using the two methods of estimating LPC parameters—they were similar but hardly identical. Next we compared how well the code books repre-

TABLE XVII
CLASSIFICATION OF RLD USING d_{GO} ON VARIOUS SPEAKER-INDEPENDENT CODE BOOKS
(r = FIXED-SIZE CODE BOOK RATE)
(u = NO. OF UTTERANCES PER SPEAKER IN TRAINING SEQUENCE)

Method	No. Class	IS Code Books			GN Code Books		
		$r = 5$ $u = 2$	$r = 5$ $u = 6$	$r = 6$ $u = 6$	$r = 5$ $u = 2$	$r = 5$ $u = 6$	$r = 6$ $u = 6$
Auto	520	77.5	77.3	78.1	74.2	77.1	78.5
Burg	520	77.9	81.9	78.6	72.5	80.4	78.7

TABLE XVIII
SPEAKER-INDEPENDENT CLASSIFICATION USING d_{GO}
ON RATE-5, CLUSTERED GN-CODE BOOKS

Speaker	No. Class	2-Utterance Training				9-Utterance Training			
		Errors	% Right	F_{av}	F_σ	Errors	% Right	F_{av}	F_σ
WMF	520	61	88.3	0.183	0.162	63	87.9	0.188	0.164
RLD	520	107	79.4	0.139	0.178	91	82.5	0.173	0.198
RGL	520	61	88.3	0.229	0.196	41	92.1	0.256	0.201
MSW	520	65	87.5	0.215	0.194	45	91.4	0.237	0.210
GRD	520	119	77.1	0.134	0.158	106	79.6	0.121	0.164
TBS	520	85	83.7	0.159	0.184	77	85.2	0.162	0.170
KAB	520	83	84.0	0.178	0.187	57	89.0	0.221	0.192
REH	520	41	92.1	0.270	0.217	18	96.5	0.321	0.254
all	4160	622	85.1	0.189	0.191	498	88.0	0.210	0.205

sented the training sequences by examining the average distortion for encoding them. We computed the code book set distortion AD by averaging the average distortion with which each code book encoded its training sequence. As in [34], we computed the RMS log spectrum error for each code book set, $SE = 6.142(e^{AD} - 1)^{1/2}$. The average difference in SE between the autocorrelation method and the Burg technique code books was 0.05 dB for IS code books and 0.06 dB for GN code books. Although the code books designed using the Burg technique had smaller average distortion, the difference appears to be too small to have a significant effect on classification accuracy.

As an additional test, we classified the 520 utterances of one female speaker (HNJ) using d_{GO} on the 4-male, rate-5 code books designed from two utterances per speaker. The classification accuracy was only 58.5 percent for the IS code books and 37.9 percent for the GN code books. These are significantly worse than the results for RLD (a male).

C. Speaker-Independent Results

We performed speaker-independent experiments using the 8 males in the TI data base. To increase the total number of classifications, we classified all 520 utterances of each male in turn using code books designed from the other seven. Except for the training sequences, parameters were the same as those in the speaker-dependent experiment discussed in Section V (Table I)—rate-5 clustered GN code books, $E_{min} = 250$, autocorrelation method of LPC analysis with Hamming windowing and 90 percent pre-emphasis, $N = W = 250$, and $M = 16$. For each speaker, we generated two GN code book sets—one from training sequences of two utterances per remaining speaker and one from nine utterances per remaining speaker. The two-utterance training sequences were subsets of the nine-utterance training sequences. We are not suggesting that multiple repetitions by the same speaker is a good way to train a speaker-independent word recognizer; we did it only to evaluate the adequacy of using only seven speakers in the training sequence. Only d_{GO} was used as a distortion measure for classification.

Table XVIII contains the results for the 8 male speakers in the TI data base. Tables XIX and XX are the confusion matrices for the two different code book sets. Accuracy is higher for the code books designed from the longer training sequences. This suggests that higher accuracy would result from a training sequence with more speakers. Comparing Tables III, XIX, and XX shows that the same types of errors occur in both the speaker-dependent and speaker-independent cases.

As in the speaker-dependent case, we obtained speaker-independent results for the subset of the vocabulary consisting only of the digits. Table XXI shows the classification results and Tables XXII–XXIII are the confusion matrices. Again, the higher accuracy of the nine-utterance per speaker code books suggests that better performance could be achieved by increasing the number of speakers in the training sequence. The results show that speaker-independent digit classification can be performed with about 95 percent accuracy, which is quite good.

VIII. COMPUTATIONAL CONSIDERATIONS

Most of the software for these experiments was written in FORTRAN-77 and run on a DEC VAX11/750 with a floating point accelerator. Generating the fixed-size, rate-5 clustered code books required 1–1.5 minutes of execution time each. Classification of a single utterance with these

TABLE XIX
SPEAKER-INDEPENDENT CONFUSION MATRIX FOR 2 - UTTERANCE PER SPEAKER TRAINING SEQUENCES

	0	1	2	3	4	5	6	7	8	9	ENTER	ERASE	GO	HELP	NO	RUBOUT	REPEAT	STOP	START	YES
0	201	·	3	1	·	·	·	·	·	·	·	·	1	·	·	1	1	·	·	·
1	1	156	·	·	1	·	·	1	·	14	·	·	1	4	26	4	·	·	·	·
2	6	·	195	·	1	·	1	·	·	·	1	·	1	·	·	1	2	·	·	·
3	1	·	·	166	·	·	·	·	·	·	3	5	·	·	·	·	33	·	·	·
4	·	2	·	·	196	4	·	·	·	·	·	·	·	·	·	4	·	·	2	·
5	·	·	1	·	·	162	·	·	·	3	·	·	·	9	1	13	·	18	2	·
6	·	·	1	·	·	·	187	1	·	·	·	2	·	·	·	·	·	3	·	14
7	·	·	·	·	·	·	3	189	·	·	·	·	·	·	·	·	·	16	·	·
8	·	·	·	12	·	·	2	·	130	·	1	5	·	·	·	·	39	·	·	19
9	·	5	·	·	·	11	·	5	·	166	3	·	·	6	3	9	·	·	·	·
ENTER	·	·	1	1	·	·	·	·	·	·	204	·	·	·	·	·	2	·	·	·
ERASE	·	·	1	2	·	·	·	·	·	·	·	203	·	·	·	·	·	·	1	1
GO	2	12	4	·	·	·	·	·	·	·	·	·	121	41	19	9	·	·	·	·
HELP	·	3	·	·	·	6	·	·	·	·	·	·	5	187	2	4	·	1	·	·
NO	·	26	·	·	1	·	·	1	·	1	·	·	14	28	134	2	·	1	·	·
RUBOUT	2	·	·	·	·	2	·	·	·	3	·	·	·	·	·	200	·	1	·	·
REPEAT	2	·	·	21	·	·	·	·	·	·	·	11	·	·	·	·	174	·	·	·
STOP	·	·	·	·	·	2	·	1	·	·	·	·	·	·	·	1	·	179	25	·
START	9	·	·	·	·	·	2	·	·	·	·	·	1	·	·	5	·	2	188	1
YES	·	·	·	·	·	·	1	·	·	·	·	6	·	·	·	·	·	1	·	200

TABLE XX
SPEAKER-INDEPENDENT CONFUSION MATRIX FOR 9-UTTERANCE PER SPEAKER TRAINING SEQUENCES

	0	1	2	3	4	5	6	7	8	9	ENTER	ERASE	GO	HELP	NO	RUBOUT	REPEAT	STOP	START	YES
0	198	·	1	·	·	·	·	1	·	·	·	3	3	·	·	1	1	·	·	·
1	·	145	·	·	1	3	·	1	·	9	·	·	·	5	41	2	·	·	1	·
2	4	·	196	·	1	·	1	·	·	·	·	·	3	·	·	1	2	·	·	·
3	1	·	·	168	·	·	·	·	·	·	1	8	·	·	·	·	30	·	·	·
4	·	·	·	·	204	·	·	·	·	·	·	·	·	1	·	1	·	·	2	·
5	·	·	·	·	·	180	·	·	·	2	·	·	·	14	·	5	·	4	3	·
6	·	·	·	·	·	·	191	1	·	·	·	6	·	·	·	·	·	·	·	10
7	·	·	·	·	·	·	·	196	·	·	·	·	·	·	·	·	·	2	·	10
8	·	·	·	·	4	·	·	·	161	·	1	7	·	·	·	·	18	·	·	17
9	·	6	·	·	·	14	·	2	·	170	7	·	·	3	4	2	·	·	·	·
ENTER	·	·	·	·	3	·	·	·	·	·	203	1	·	·	·	·	1	·	·	·
ERASE	·	·	·	·	1	·	·	·	·	·	·	207	·	·	·	·	·	·	·	·
GO	3	·	1	·	·	·	·	·	·	·	·	·	134	45	16	9	·	·	·	·
HELP	·	1	·	·	·	10	·	·	·	·	·	·	1	192	·	2	·	1	1	·
NO	2	24	·	·	1	·	·	1	·	·	·	·	8	26	143	1	·	2	·	·
RUBOUT	·	·	·	·	·	2	·	1	·	3	·	·	·	·	·	199	·	2	1	·
REPEAT	·	·	·	7	·	·	·	·	1	·	1	5	·	·	·	·	194	·	·	·
STOP	·	·	·	·	·	2	·	·	·	·	·	·	·	·	·	2	·	176	28	·
START	·	·	·	·	·	1	·	1	·	·	·	1	·	·	·	1	·	·	202	2
YES	·	·	·	·	·	·	1	·	·	·	·	4	·	·	·	·	·	·	·	203

TABLE XXI
SPEAKER-INDEPENDENT CLASSIFICATION OF DIGITS USING d_{GO} ON RATE-5, CLUSTERED GN-CODE BOOKS

Speaker	No. Class	2-Utterance Training				9-Utterance Training			
		Errors	% Right	F_{av}	F_{σ}	Errors	% Right	F_{av}	F_{σ}
WMF	260	20	92.3	0.317	0.223	18	93.1	0.320	0.221
RLD	260	7	97.3	0.283	0.197	6	97.7	0.339	0.224
RGL	260	9	96.5	0.398	0.250	6	97.7	0.408	0.274
MSW	260	20	92.3	0.360	0.259	12	95.4	0.391	0.279
GRD	260	31	88.1	0.252	0.194	22	91.5	0.252	0.215
TBS	260	10	96.2	0.330	0.229	9	96.4	0.317	0.200
KAB	260	14	94.6	0.309	0.197	11	95.8	0.378	0.228
REH	260	15	94.2	0.409	0.282	2	99.2	0.499	0.327
all	2080	126	93.9	0.332	0.236	86	95.9	0.363	0.259

TABLE XXII
SPEAKER-INDEPENDENT, DIGIT-SUBSET CONFUSION MATRIX FOR 2-UTTERANCE PER SPEAKER TRAINING SEQUENCES

	0	1	2	3	4	5	6	7	8	9
0	203	1	3	1	·	·	·	·	·	·
1	1	188	·	·	1	1	·	1	·	16
2	6	1	199	·	1	·	1	·	·	·
3	3	·	·	204	·	·	·	·	1	·
4	·	3	·	·	200	5	·	·	·	·
5	·	6	·	·	·	192	·	2	·	8
6	·	·	1	·	·	·	202	5	·	·
7	·	·	·	·	·	·	3	205	·	·
8	1	·	·	21	·	·	5	·	181	·
9	·	5	·	·	·	16	·	7	·	180

TABLE XXIII
SPEAKER-INDEPENDENT, DIGIT-SUBSET CONFUSION MATRIX FOR 9-UTTERANCE PER SPEAKER TRAINING SEQUENCES

	0	1	2	3	4	5	6	7	8	9
0	203	·	1	2	·	·	·	2	·	·
1	·	191	·	·	2	4	·	1	·	10
2	6	·	199	·	2	·	1	·	·	·
3	2	·	·	205	·	·	·	·	1	·
4	·	1	·	·	206	1	·	·	·	·
5	·	1	·	·	·	198	·	1	·	8
6	·	·	·	1	·	·	205	2	·	·
7	·	·	·	·	·	·	2	206	·	·
8	2	·	·	5	·	·	3	·	198	·
9	·	8	·	·	·	15	·	2	·	183

code books took about 1 s/code book. All of the software was designed for research purposes; we are confident that specially designed programs would run considerably faster.

In general, let N = code book size, M = LPC model order, and L = length of an input utterance in frames. Then GN code books each require $N(M + 1)$ storage locations, and the classification of an input utterance requires NL distortion calculations per code book. Several approaches could be used to reduce these requirements. For example, some codewords might contribute more to incorrect classifications than to correct classifications. If they can be found and removed, code book storage would decrease and classification speed would increase. Also, one could track the accumulating average distortions during classification, and reject some of the hypotheses without having to compute the average distortion of every code book over the entire input utterance.

Even without such improvements, it is instructive to compare the computational requirements of the vector quantization (VQ) approach with those of DTW. Both requirements are dominated by the number of distortion calculations. In typical DTW approaches, the reference template and the input utterance are linearly normalized to the same length L before performing DTW, and appropriate constraints are applied to the search path [5]. High recognition accuracies can then be achieved with αL^2 distortion computations per reference template, where α is in the approximate range 0.2 to 0.3 [5]. Thus the ratio P of the number of distortion calculations required by the VQ approach to the number required by the DTW approach is about $P \approx N/\alpha L$. For fixed-rate code books with $N = 2^R$, and for a nominal value of $\alpha \approx 0.25$, the ratio becomes $P \approx 2^{R+2}/L$. We shall assume that a typical input utterance is $L \approx 32$ frames long (640 ms at 20 ms per frame)—this is perhaps too large, but it is conveniently a power of two. It follows that the ratio of distortion calculations becomes

$$P \approx 2^{R-3}. \tag{13}$$

For our best results—achieved with rate-5 code books—(13) shows that DTW requires fewer distortion calculations. But for rate-2 and rate-3 code books, which still achieve excellent recognition accuracies of about 98 percent (see Table VII), the VQ approach requires fewer or about the same number of distortion calculations.

A DTW approach using ordered, graph-searching techniques can reduce the number of distortion computations by an additional factor of about 2.5 [35], which changes the exponent in (13) from $R - 3$ to about $R - 2$. It does so, however, at the expense of a more complicated control structure. Whether the overall computational requirement is reduced depends strongly on the hardware available for distortion calculations [35].

The foregoing comparison does not apply to the speaker-independent case, since conventional speaker-independent recognition systems usually have several templates for each word in the recognition vocabulary. Since the VQ approach still requires only one code book, considerably fewer distortion calculations are required for the VQ approach than for DTW. The moderately good performance of the VQ approach in this case suggests its use as a preprocessor for DTW systems [14].

During classification, the input speech frames provide the argument f in (7), (9), or (10). It follows that both the time domain autocorrelations $r(n)$ and the LPC gain squared σ^2 must be known for each input frame, which in turn means that an LPC analysis must be performed. For d_{IS} and d_{GO}, however, the gain enters as a constant term $(\ln(\sigma^2))$ that contributes a constant term in the computation of the average code book distortions (3). The classification can therefore be performed without this term, which means than no LPC analysis of the input utterance is required. However, because only the relative values of the classification distortion will be known in this case, an overall rejection threshold cannot be used.

IX. Conclusion

The strongest conclusion from our study is that, for speaker-dependent recognition of a 20-word vocabulary, 99 percent accuracy results from classification using the average distortion of specially designed vector quantization code books. For the digit subset, performance is considerably better. This performance is about the same as that achieved by DTW, which shows that much more can be done without time-sequence information than is commonly assumed. Other strong conclusions are as follows:

a) The method achieves almost 98 percent accuracy with rate-2 code books—only four codewords per code book.

b) The method achieves 88 percent accuracy for speaker-independent recognition of the 20-word vocabulary and 95 percent accuracy for the digit subset.
c) Clustered code books should be designed to fixed-rate rather than fixed-distortion criteria.
d) With 1-utterance unclustered code books, the method achieves about 90 percent accuracy for speaker-dependent recognition of the 20-word vocabulary and 95 percent accuracy for the digit subset. This approach has the advantages of fast training and no requirement for code book design software.

For suitably chosen vocabularies, our results show that characteristic spectra contain enough information for recognition, and that information-theoretic clustering does a good job of extracting that information from training data.

Given the performance obtained without exploiting time sequence information, it seems worthwhile to consider ways of increasing performance by incorporating some time sequence information. We see two basic approaches. One is to observe the sequence of code words that are closest to frames of an input utterance when the utterance is classified with the correct code book. If typical code book "trajectories" can be defined, it might be possible to incorporate them into the classification algorithm. Another approach is to divide utterances into sections and to use a separate code book for each section, an approach suggested by Buzo [13]. Preliminary experiments with this approach show substantial improvements in both classification accuracy and computational efficiency [36].

ACKNOWLEDGMENT

We thank J. Buck, A. Buzo, R. Johnson, and L. Rabiner for helpful discussions. We thank J. Buck for writing some of the software and for providing various additional assistance. We thank R. M. Gray for providing some of the vector quantization software and for reviewing an earlier draft of this paper, and we thank T. Schalk for his help in obtaining the data base. We also benefited from reviews by two anonymous referees.

REFERENCES

[1] L. R. Rabiner and S. E. Levinson, "Isolated and connected word recognition-theory and selected applications," *IEEE Trans. Commun.*, vol. COM-29, pp. 621–659, 1981.
[2] G. R. Doddington and T. B. Schalk, "Speech recognition: Turning the ory to practice," *IEEE Spectrum*, pp. 26–32, Sept. 1981.
[3] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 67–72, Feb. 1975.
[4] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 43–49, Feb. 1978.
[5] C. Myers, L. R. Rabiner, and A. E. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-28, pp. 623–635, Dec. 1980.
[6] A. Buzo, A. H. Gray Jr., R. M. Gray, and J. D. Markel, "Speech coding based upon vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 562–574, Oct. 1980.
[7] R. M. Gray, A. H. Gray, Jr., G. Rebolledo, and J. E. Shore, "Rate-distortion speech coding with a minimum discrimination information distortion measure," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 708–721, Nov. 1981.
[8] J. E. Shore and R. M. Gray, "Minimum-cross-entropy pattern classification and cluster analysis," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. PAMI-4, pp. 11–17, Jan. 1982.
[9] R. M. Gray, A. Buzo, A. H. Gray, Jr., and Y. Matsuyama, "Distortion measures for speech processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 367–376, Aug. 1980.
[10] J. E. Shore and D. Burton, "Discrete utterance speech recognition without time normalization," in *Proc. ICASSP82*, Rep. IEEE 82CH1746-7, pp. 907–910, May 1982.
[11] J. E. Shore and D. Burton, "Discrete utterance speech recognition without time normalization—Recent results," in *Proc. 1982 Int. Conf. Pattern Recognition*, Rep. IEEE 82CH1801-0, pp. 582–584, Oct. 1982.
[12] R. Hamabe, Y. Yamada, M. Murata, and T. Namekawa, "A speech recognition system using inverse filter matching technique," in *Proc. Ann. Conf. Inst. of Television Engineers*, Kyushu University, June 1981, (in Japanese).
[13] H. G. Martinez, C. Riviera, and A. Buzo, "Discrete utterance recognition based upon source coding techniques," in *Proc. ICASSP82*, Rep. IEEE 82CH1746-7, pp. 539–542, May 1982.
[14] L. R. Rabiner, Bell Laboratories, private communication.
[15] F. Itakura and S. Saito, "Analysis synthesis telephony based on the maximum likelihood method," in *Reports of the 6th Int. Cong. Acoustics*, 1968.
[16] J. E. Shore, "Minimum cross-entropy spectral analysis," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 230–237, Apr. 1981.
[17] S. Kullback, *Information Theory and Statistics*. New York: Dover, 1969.
[18] J. E. Shore and R. W. Johnson, "Properties of cross-entropy minimization," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 472–482, July 1981.
[19] ——, "Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 26–37, Jan. 1980.
[20] R. W. Johnson and J. E. Shore, "Minimum-cross-entropy spectral analysis of multiple signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, June 1983.
[21] ——, "Multi-signal minimum-cross-entropy spectrum analysis with weighted priors," NRL Report in publication, Naval Research Laboratory, Washington, D.C., 1983. (Submitted to *IEEE Trans. Acoustics, Speech, Signal Processing.*)
[22] R. W. Johnson, J. E. Shore, D. Burton, and J. Buck, "Speech noise reduction by means of multi-signal minimum cross-entropy spectral analysis," in *Proc. ICASSP83*, Rep. IEEE 83CH1841-6, pp. 1129–1132, April 1983.
[23] J. E. Shore, "Information theoretic approximations, for M/G/1 and G/G/1 queuing systems," *Acta Informatica*, vol. 17, pp. 43–61, 1982.
[24] S. P. Lloyd, "Least squares quantization in PCM," Bell Laboratories Tech. Rep., Murray-Hill, NJ, 1957.
[25] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.
[26] L. R. Rabiner and M. Sambur, "An algorithm for determining the endpoints of isolated utterances," *Bell. Syst. Tech. J.*, vol. 54, pp. 297–315, Feb. 1975.
[27] L. Lamel *et al.*, "An improved endpoint detector for isolated word recognition," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-29, pp. 777–785, Aug. 1981.
[28] R. E. Crochiere and L. R. Rabiner, "Interpolation and decimation of digital signals—A tutorial review," in *Proc. IEEE*, vol. 69, pp. 300–331, Mar. 1981.
[29] T. E. Tremain, "The government standard linear predictive coding algorithm: LPC-10," *Speech Tech.*, vol. 1, pp. 40–49, April 1982.
[30] L. Miclet and A. Nehame, "Experience a reconnaissance de la parole par prediction linear," Dept. Systeme et Communication Rep. ENST-C-79020, ENST, Sept. 1979.
[31] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math., Stat.,*

and Prob., vol. 1, Univ. California Press, pp. 281–286, 1967.
[32] P. E. Hart, "The condensed nearest neighbor rule," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 515–516, 1968.
[33] S. Roucos, R. Schwartz, and J. Makhoul, "Segment quantization for very-low-rate speech coding," in *Proc. ICASSP82*, Rep. IEEE 82CH1746-7, pp. 1565–1568, May 1982.
[34] G. Rebolledo, R. M. Gray, and J. P. Burg, "A multirate voice digitizer based upon vector quantization," *IEEE Trans. Commun.*, vol. COM-30, pp. 721–727, April 1982.
[35] M. K. Brown and L. R. Rabiner, "An adaptive, ordered, graph search technique for dynamic time warping for isolated word recognition," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-30, pp. 535–544, Aug. 1982.
[36] J. E. Shore, D. Burton, and J. Buck, "A generalization of isolated word recognition using vector quantization," in *Proc. ICASSP83*, Rep. 83CH1841-6, pp. 1021–1024, April 1983.

Minimax Optimal Universal Codeword Sets

PETER ELIAS, FELLOW, IEEE

Abstract—**In an interactive multi-user data-processing system, a user knows the probabilities of his messages and must encode them into a fixed system-wide variable-length codeword set. He needs to receive the answer to his last message before selecting the next, so his encoding is one-shot. To minimize average codeword length he encodes his messages in order of decreasing probability into codewords in order of increasing length. An algorithm is given which, for each of several measures of performance, finds the codeword set best by that measure for the worst user, and some of the minimax optimal codeword sets the algorithm has found. Some of the results hold for all user distributions: others require, e.g., that all users send exactly or at most m distinct messages, or that there is an integer k such that no user has a message of probability greater than $1/k$.**

I. Introduction

In an interactive multi-user data-processing system each user or user group may have a different message set or probability distribution, but it may be convenient for the system to require that each user encode his messages into a fixed systemwide set of codewords. Since a user may need to receive the answer to his last message before sending the next, his encoding must be one-shot. To minimize average codeword length he assigns his messages in order of decreasing probability to codewords in order of increasing length. He evaluates the cost of using the system by comparing the resulting average codeword length to the average length of a set of codewords designed to be optimal for his particular probability distribution.

As an example, let a user whose probability distribution has entropy H bits per message encode his kth most probable message into the standard binary representation of the integer k (of length $1 + \lfloor \log_2 k \rfloor$, where $\lfloor x \rfloor$ is the largest integer no greater than x) prefixed by a sequence of $\lfloor \log_2 k \rfloor$ 0's, so that the most probable codeword is 1 and the seventh most probable codeword is 00111. It is shown in [2] that the resulting average codeword length is no more than $1 + 2H$, and therefore cannot be much more than twice as great as the average length of the best (Huffman) code for that distribution, which is known to lie between H and $1 + H$.

We discuss in this paper the problem of finding the best codeword set in a minimax sense to use in such a system—i.e., the codeword set which minimizes over all admissable codeword sets the maximum (over some class of acceptable user probability distributions) of some measure of the extra cost of using that single codeword set for all distributions in the class. The solution to the problem depends on the class of acceptable user probability distributions and on the cost measure used. Section II establishes notation for distributions, entropies and codeword lengths and gives some well-known single-user results for reference. Section III establishes notation for the multiuser problem and defines a set of relevant cost measures and a number of interesting classes of probability distributions. Section IV summarizes previous work [1]–[4] in terms of these definitions. Section V shows that for each of the cost measures and classes of distributions defined in Section III the worst distributions are uniform, and that for the cost measures most relevant for comparing one-shot encodings, the worst uniform distributions lie in a subset whose size grows only logarithmically with the maximum allowed number of messages. These results make it computationally feasible to check a proposed codeword set for optimality. Section VI uses the results in Section V to derive an

Manuscript received January 13, 1982; revised July 29, 1982. This work was supported in part by the Army Research Office under Contract DAAG29-77-C-0012.

The author is with the Department of Electrical Engineering and Computer Science and the Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139.

0018-9448/83/0700-0491$01.00