Techniques for Information Retrieval from Speech Messages

R.C. Rose

The goal in speech-message information retrieval is to categorize an input speech utterance according to a predefined notion of a topic, or message class. The components of a speech-message information-retrieval system include an acoustic front end that provides an incomplete transcription of a spoken message, and a message classifier that interprets the incomplete transcription and classifies the message according to message category. The techniques and experiments described in this paper concern the integration of these components, and represent the first demonstration of a complete system that accepts speech messages as input and produces an estimated message class as output. The promising results obtained in information retrieval on conversational speech messages demonstrate the feasibility of the technology.

THE GOAL IN SPEECH-MESSAGE information retrieval is similar to that of the more well-known problem of information retrieval from text documents. Text-based information-retrieval systems sort large collections of documents according to predefined relevance classes. This discipline is a mature area of research with a number of well-known document-retrieval systems already in existence. Speech-message information retrieval is a relatively new area, and work in this area is motivated by the rapid proliferation of speech-messaging and speech-storage technology in the home and office. A good example is the widespread application of large speech-mail and speech-messaging systems that can be accessed over telephone lines. The potential length and number of speech messages in these systems make exhaustive user review of all messages in a speech mailbox difficult. In such a system, speech-message information retrieval could automatically categorize speech messages by context to facilitate user review. Another application would be to classify incoming customer telephone calls automatically and route them to the appropriate customer service areas [1].

Unlike information-retrieval systems designed for text messages, the speech-message information-retrieval system illustrated in Figure 1 relies on a limited-vocabulary acoustic front end that provides only an incomplete transcription of a spoken message. The second stage of the system is a message classifier that must interpret the incomplete transcription and classify the message according to message category. In our system the acoustic front end is based on a hidden-Markov-model (HMM) word spotter [2]. The techniques described in this paper concern the design and training of the second-stage message classifier and the integration of the message classifier with the acoustic front end [3]. The major result described in the paper is the demonstration and evaluation of an experimental system for speech-message information retrieval in which speech messages are automatically categorized into message classes.

The problem of speech-message information retrieval must be distinguished from that of speech-message understanding. In speech-message understanding, an utterance is analyzed at acoustic, syntactic, and semantic levels to provide a complete description of the utterance at all levels. Determining a complete description, however, is a difficult problem, especially for the unconstrained

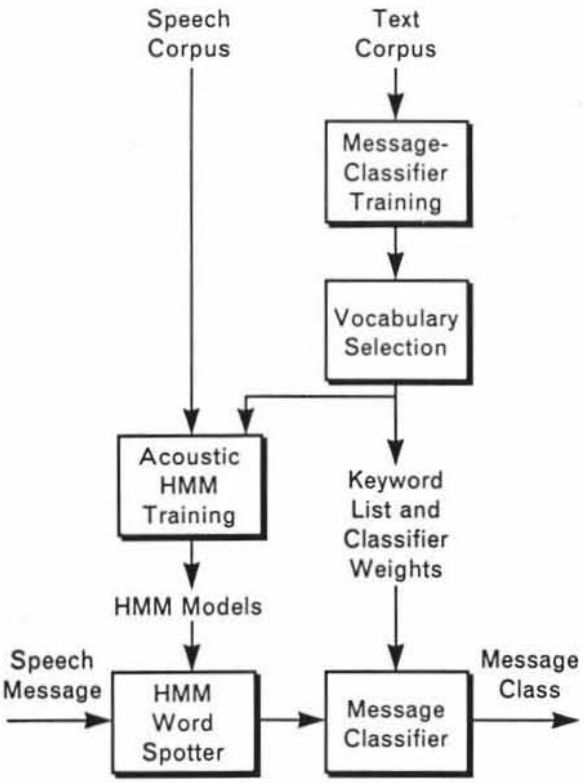

FIGURE 1. Block diagram of a speech-message information-retrieval system. The hidden-Markov-model (HMM) word spotter accepts a continuous-speech utterance as input and produces a partial transcription of the utterance according to a predefined keyword vocabulary. The message classifier accepts the speech message in this reduced form and assigns it to a message class.

conversational speech messages described in the following section. The goal in speech-message information retrieval is more modest; such a system attempts only to extract the most general notion of topic or category from the message. The purpose of this paper is to demonstrate the feasibility of a speech-message information-retrieval system.

In configuring the system to a particular task, we assume that both speech and text corpora exist that represent the speech messages in each message category. While the speech corpus is used for training statistical hidden Markov acoustic models for the word spotter, the text corpus, which contains text transcriptions of speech messages, is used for training the second-stage message classifier. The next section describes the speech-message classification task, along with the speech and text corpora used to define the task.

The automatic techniques and experiments for speech-message information retrieval are described in two parts. First, a perfect acoustic front end is assumed, and the attention is focused on the message classifier. The section entitled "Message Classification" describes the message classifier model and the techniques used for training this model. Results of message classification from text transcriptions of speech messages are also presented. The second part of the paper concerns the complete problem of information retrieval from speech messages. The section entitled "Information Retrieval from Speech Messages" describes the acoustic word spotter and techniques for integrating the acoustic front end with the second-stage message classifier. Results are presented for both word-spotting performance and information-retrieval performance from speech messages.

Speech-Message Information Retrieval

The most difficult problem in performing a study on speech-message information retrieval is defining the task. The definition of a message class is a difficult issue, and the relevance of a particular speech message with respect to a message class cannot always be determined with certainty. We were fortunate in that a speech corpus already existed that was suitable for this study. In this

Table 1. Message-Classification Performance on Text Transcriptions of Speech Messages

Initial Message-Classification Performance (240 Words)		
Message Class	Percent Correct Train	Test
Toy Description	92.1	86.5
Abstract Object Description	94.1	74.5
General Discussion	84.0	68.0
Road Rally Map Reading	100.0	100.0
Photographic Interpretation	88.2	86.5
Cartoon Description	96.0	80.7
Overall	91.3	81.3

speech corpus, natural conversation was elicited from speakers by having them interact with an interlocutor in performing a number of different scenarios. The speech messages used in this study were excerpted from these conversations, and the associated scenarios were used to define the message classes. Hence we avoided the difficult problem of defining the message categories and obtaining relevant speech messages for those categories by defining the categories according to the scenarios used to collect the speech messages.

The corpus consists of 510 conversational speech messages from 51 speakers; each message is at least 30 seconds in length. Each message is orthographically transcribed, and the entire set of messages consists of approximately 40,000 words. The messages were collected under six different scenarios, which are listed in Table 1. Most of the scenarios are relatively unconstrained and require the speaker to describe several items to the interlocutor. For example, in the photographic interpretation scenario the speaker has a collection of black-and-white photographs, and is asked to describe each one to the interlocutor in a few sentences. When compared to text corpora (containing millions of words) that are used to train statistical language models for large-vocabulary continuous-speech recognizers, this amount of text is extremely small. The interest in this work, however, is in developing systems that can be easily reconfigured for a speech-message information-retrieval task. We are interested in determining whether techniques can be developed that can profit from a more modest representation of the domain of interest.

Message Classification

This section presents techniques for configuring the message-classification portion of the speech-message information-retrieval system. The experiments described in this section present the best-case speech-message information-retrieval performance by using text transcriptions of speech messages to evaluate the message-classification techniques. The first step in training the message classifier involves training the weights for the message classifier. The second step is to choose a subset of the total words in the text corpus to use as the message-classification vocabulary. This second step is referred to below as the process of vocabulary selection.

Message-Classifier Model

Figure 2 shows the message-classifier model used in these experiments. An input message M is assumed to be a collection of L independent words. We also assume that there exists a set $\mathbf{V} = \{w_1, \ldots, w_K\}$ consisting of K words that forms the message-classification vocabulary. For each vocabulary word there exists a message-classifier activation $s_k(\mathbf{M})$ that is activated by the occurrence of vocabulary word w_k in the input message, so that $s_k(\mathbf{M}) = n$ if there are n occurrences of word w_k in message M. The subsection below, "Message-Class Corrective Keyword Detection," describes a more interesting mapping of input word to message-classifier activation; this mapping reflects the degree of confidence in the detection of that keyword from the input speech.

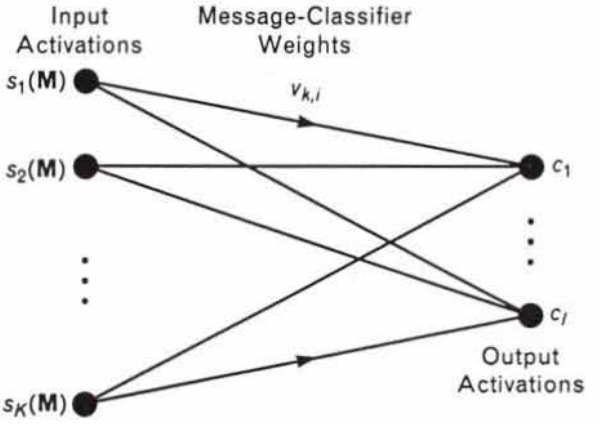

FIGURE 2. Message-classifier model. The message classifier assumes that the input message **M** consists of a set of independent words. The message-classifier weights $v_{k,i}$ are trained by using a mutual information criterion. The message-classifier output activations c_i represent the estimate of the log probability of a message class $\mathbf{C}_i$, given the input message **M**.

Modeling a message as a set of independent words, or *unigrams*, as opposed to using models based on higher-order statistics, was motivated largely by the relatively small training corpus. Even though we investigated anecdotal examples of messages formed by examining co-occurrence relationships among words, estimating the statistics of these word co-occurrence relationships was difficult with such a small amount of training data.

The output of the classifier corresponds to the message-class activations $c_1, \ldots, c_I$. The problem of message-

classifier training corresponds to determining the network weights $v_{k,i}$ on the basis of training messages from all message classes. The approach taken here is motivated by J.B. Haseler [4] and was also taken by A.L. Gorin et al. [1].

For a simple two-class message classifier, the weights are chosen to maximize the log likelihood of the *interesting* message class relative to the *uninteresting* class, given an input message. If the words that form the message are assumed independent, and the message classes are assumed to be equally probable, then this likelihood is expressed as

$$\log \frac{P(\mathbf{C}_1|\mathbf{M})}{P(\mathbf{C}_2|\mathbf{M})} = \log \frac{P(\mathbf{M}|\mathbf{C}_1)P(\mathbf{C}_1)}{P(\mathbf{M}|\mathbf{C}_2)P(\mathbf{C}_2)} = \sum_{w_k \in \mathbf{V}} \log \frac{P(w_k|\mathbf{C}_1)}{P(w_k|\mathbf{C}_2)}, \quad (1)$$

where the sum is over all words in the message that are contained in the message-classification vocabulary. For the general I-class message-classification problem the above expression can be generalized so that the weights are chosen to maximize the *a posteriori* log probability of a message class. Again, if we assume independent words in a message, we can show that

$$\log P(\mathbf{C}_i|\mathbf{M}) = \sum_{w_k \in \mathbf{V}} \log \frac{P(\mathbf{C}_i, w_k)}{P(\mathbf{C}_i)P(w_k)} + \log P(\mathbf{C}_i) + \sum_{w_k \in \mathbf{V}} \log \frac{P(w_k)}{P(\mathbf{M})} \quad (2)$$

for each class $\mathbf{C}_1, \ldots, \mathbf{C}_I$. For the two-class case, the message-classifier output for the ith topic class is given as c_i, where

$$c_i = \sum_{w_k \in \mathbf{V}} v_{k,i} s_k(\mathbf{M}).$$

For this case, the weight $v_{k,i}$ between the word w_k and the topic c_i is

$$v_{k,i} = \log \frac{P(w_k|\mathbf{C}_1)}{P(w_k|\mathbf{C}_2)},$$

which is the conditional information of w_k. For the general I-class case, the message classifier output c_i is

$$c_i = \sum_{w_k \in \mathbf{V}} v_{k,i} s_k(\mathbf{M}) + \log P(\mathbf{C}_i),$$

where

$$v_{k,i} = \log \frac{P(\mathbf{C}_i, w_k)}{P(\mathbf{C}_i)P(w_k)}, \quad (3)$$

which corresponds to the mutual information between message class $\mathbf{C}_i$ and word w_k.

Estimating Message-Classifier Weights

The message-classifier weights given in Equation 3 are used directly in the classifier illustrated in Figure 2. This choice of weights is optimal only under the assumption that the probabilities in Equation 3 are homogeneous across all messages in a message class. This assumption is generally not the case for the moderate-length messages described in the previous section. An alternative means for training the classifier would be to learn the weights by minimizing a function of the overall message-classification error. This method is not possible, however, because the vocabulary V is not known in advance.

The weights in Equation 3 are obtained directly by estimating the probabilities in the equation from sample statistics derived from the frequency of occurrence of the individual words in text messages. Several steps precede the estimation of these sample statistics. The first step is the removal of frequently occurring common words from the messages. Second, noun plurals and verb tenses are removed by reduction to a common baseform through a set of word-stemming rules. Finally, word counts are accumulated and used to estimate the probabilities in Equation 1 and Equation 2. Estimating these probabilities requires special precautions because, even for an extremely large sample of training text, important words occur infrequently. We use the Turing-Good estimate of word frequency to overcome the problem of estimating probabilities of words that occur infrequently in the training text [5]. This estimate assumes that each word w_k is binomially distributed in the text corpus, and has the overall effect of increasing the probability of words that occur infrequently in the training text, while decreasing the probability of more frequently occurring words.

Vocabulary Selection

Earlier we estimated a set of message-classifier weights for all of the words in the text corpus. In this section, we investigate techniques for choosing a smaller subset of the total words in the corpus for use as a message-classification vocabulary. This process is referred to as *vocabulary selection.* The goal in vocabulary selection is to reduce the size of the message-classification vocabulary while maintaining an acceptable level of message-classification performance.

Vocabulary selection is motivated by three issues. The first issue relates to the reduction in computational complexity of the full speech-message information-retrieval system. The second issue is concerned with the incorrect assumptions used in motivating the weight-estimation procedure described earlier. Clearly, as average message lengths become shorter, the probabilities of words estimated from the entire training corpus become less and less representative of the probabilities of words appearing within individual messages. To deal with this issue, we chose a set of vocabulary words to minimize a function of the overall message-classification error. The third and most subtle issue relates to the independence assumption in which potential interactions among vocabulary words are ignored. Often a particular word on its own carries little information to discriminate one message class from another; when considered in the context of other words, however, the word can become an important discriminant.

Vocabulary selection is addressed here as a feature-selection problem. Exhaustive evaluation of all possible combinations of vocabulary words is not practical because the number of possible word combinations grows exponentially with the number of words. Two different suboptimal feature-selection techniques, known as genetic-algorithm (GA) search and heuristically guided sequential search, were investigated. Both techniques were evaluated in terms of their ability to obtain message-classification vocabularies that maximize message-classification performance.

GA search is a form of directed random search that successively recombines the most-fit members of a population (the fitness of a member of a population is defined by the evaluation of a predefined fitness function). The goal is to create new members of the population with increasing levels of fitness. The members of the population are strings of bits, and each bit in the string enables or disables the use of a corresponding feature. In feature selection, the fitness function is the percent-correct classification performance of the resulting feature set.

In the vocabulary-selection problem, a feature corresponds to a vocabulary word, so each member of the population in the GA search corresponds to a different message-classification vocabulary. The fitness function for a particular member of the population corresponds to the message-classification performance of the corresponding vocabulary. In the vocabulary-selection experiments described above, the GA search finds a smaller subset of an N-word vocabulary without sacrificing message-classification performance. This process is accomplished by defining each member of the population as an N-bit string. A smaller subset of the original N-word vocabulary is obtained by enabling those bits which correspond to a subset of the total vocabulary.

To stimulate the reduction of vocabulary size in GA search, a bonus must be applied to the fitness function to reward those strings with a smaller number of vocabulary words. This bonus function is a constant multiplied by the number of vocabulary words not used, and is added to the fitness value for the string. Vocabulary reduction must not be obtained, however, at the expense of message-classification accuracy. To prevent the loss of accuracy, the bonus is applied only to strings whose classification accuracy is as good as or better than any previous string. A large number of alternative strategies are available to regulate the often conflicting requirements of classification accuracy, vocabulary size, and convergence rate. These alternatives are discussed in Reference 6.

The first feature-selection technique applied to vocabulary selection is heuristically guided sequential search. Forward and backward sequential-search procedures successively add and discard individual features, respectively, on the basis of their effect on classification performance. At the kth stage of forward sequential search, all feature combinations that include the current $(k-1)$-dimensional feature vector and one of the remaining features are evaluated, and the best combination is chosen as the k-dimensional feature vector. At the kth stage of backward sequential search, a single feature of the k-dimensional feature vector that results in the smallest reduction in classification performance is discarded.

Experiments

Initial message-classification experiments were performed on the text transcriptions of the speech messages from the corpus defined for this study. Half of the transcribed messages were designated as the training dataset and the other half were designated as the test dataset. The initial vocabulary consisted of 240 words chosen by selecting 40 words for each class C_i with the highest mutual information $I(C_i,w_k)$. This initial vocabulary selection was performed on the half of the messages designated as the training dataset. Message-classifier weights were estimated from all 510 text messages. Message-classification performance was evaluated on the half of these messages designated as a test dataset by using a leave-one-out procedure. For each test message, the word frequencies for that message were subtracted from the total word-frequency count, the message-classifier weights were reestimated, and the message was labeled according to message class by using the updated classifier weights.

Table 1 gives the message-classification performance by message class for the above experiment. An overall classification performance of 81.3 percent correct was obtained with considerable variability across message classes. As might be expected, message classes corresponding to highly constrained tasks such as map reading resulted in high message-classification accuracy, while less constrained tasks such as general conversation resulted in poorer performance.

The performance of both GA search and sequential search for vocabulary selection were also evaluated on the full database described in an earlier section. The initial 240-word vocabulary described above was used as the maximum vocabulary size for both the GA and sequential search. The fitness function for all vocabulary selection procedures corresponded to the percent-correct message classification on the designated training dataset, and the performance of each procedure was evaluated on the designated test dataset.

Table 2 gives the comparative performance of these techniques. With the exception of the first row of the table, all results are reported for a 126-word vocabulary. Message-classification performance on the text messages used for vocabulary selection is given in the second column of the table, and performance on an independent test set is given in the third column. The first row of the table summarizes the overall message-classification performance of the 240-word vocabulary system as already shown in Table 1. Table 2 shows that the GA search procedure identified a 126-word vocabulary with no sacrifice in message-classification performance over the initial 240-word vocabulary. The table also shows that the GA search outperforms both the forward and backward sequential-search procedures with the same size vocabulary.

Information Retrieval from Speech Messages

This section describes a complete system for speech-message information retrieval and presents the results for this system on the speech-message information-retrieval task discussed earlier. We have already described the message classifier and the means for vocabulary selection, so now we describe the two remaining components of

Table 2. Comparison of Vocabulary-Selection Techniques

Reduced Vocabulary Performance			
Vocabulary Selection	Words	Train (%)	Test (%)
Max $I(C_i,w_k)$	240	91.3	81.3
Max $I(C_i,w_k)$	126	83.2	71.1
Forward Sequential Search	126	88.9	73.0
Backward Sequential Search	126	87.6	72.2
GA Search	126	89.2	78.6

the speech-message information-retrieval system shown in Figure 1. The first component, the HMM word spotter, is described and evaluated separately on the conversational speech messages. The second component automatically integrates the acoustic front end and the second-stage message classifier. This component, which is included in the message classifier to account for acoustic keyword confusion in the word spotter, is referred to below as a *message-class corrective keyword detection.* It compensates for the effect of keyword false alarms on performance. This second component is described and its effect on complete end-to-end speech-message information-retrieval performance is evaluated.

Hidden-Markov-Model Word Spotter

The word spotter is based on a statistical HMM representation of speech. HMMs have found wide application in speech recognition, and are the subject of many excellent review articles [7, 8]. If viewed generatively, an HMM consists of a hidden sequence of states resulting from a finite-state Markov process that is transformed into a set of observations through a set of observation densities. When HMM methods are applied to the training of a word spotter or speech recognizer, the parameters of an HMM can be estimated from example utterances to represent a particular word or subword unit. A word is spotted within a portion of an input utterance when the HMM corresponding to that word is the most likely model to have generated that portion of the utterance.

Word spotting and continuous-speech recognition (CSR) are similar problems; both involve identifying a finite vocabulary of words in continuous utterances. Word spotting, however, differs from CSR in two important aspects. The first major difference lies in the assumptions that are made about the words in the input utterance. The CSR method generally assumes that all speech presented to the recognizer consists of a finite set of vocabulary words. The word spotter must be able to accept as input completely unconstrained speech utterances that include both in-vocabulary keyword speech as well as out-of-vocabulary non-keyword speech. The second difference between word spotting and CSR is found in the mode of interaction that is generally assumed for the speaker in the two different types of systems. Most CSR systems can only interpret utterances that conform to a restrictive syntax, thus confining the user to a rigid mode of human-machine interaction. Word spotting, on the other hand, assumes that input speech can arise from completely unconstrained human-machine or even human-human interaction.

To deal with the non-keyword speech that is presented to the word spotter, we added acoustic filler models to the word spotter's vocabulary. Filler models are intended to act as explicit models of non-keyword speech, and serve as a self-normalizing threshold to disambiguate keyword speech from non-keyword speech. After experimenting with several different types of filler models, we obtained the best trade-off between performance and computational complexity when fillers were trained as

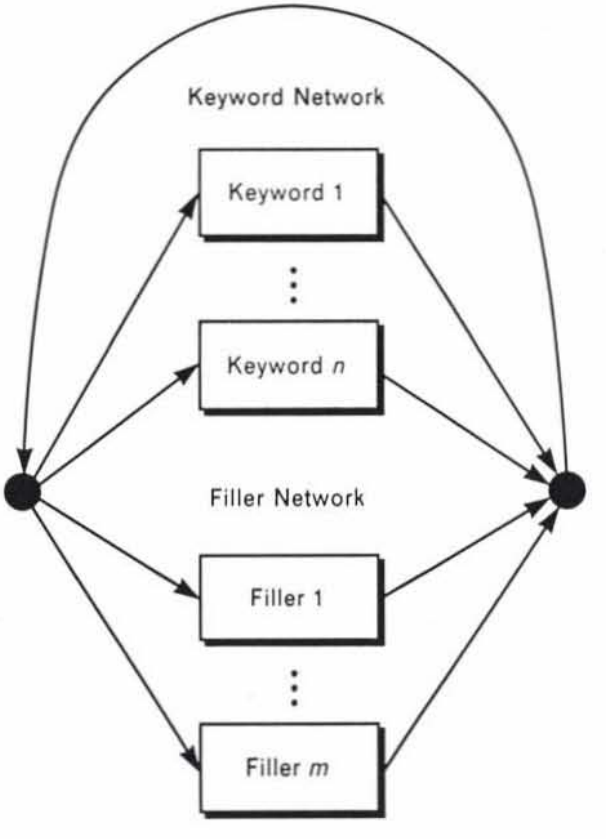

FIGURE 3. Null-grammar word-spotter network. Both keyword and fillers are represented as labeled arcs in the network. The grammar in this context is a set of rules that defines the relationship of the words and fillers in the network. A null grammar is a degenerate case that allows all words and fillers to follow one another in any sequence.

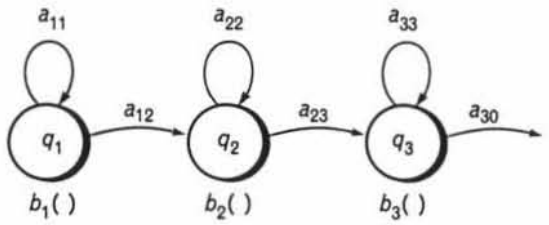

FIGURE 4. A three-state left-to-right hidden Markov subword model. The finite-state Markov chain is characterized by the transition probabilities $a_{i,j}$; the manner in which observations are generated from a state sequence is characterized by multivariate normal observation densities $b_i(\,)$.

HMMs of general-context phonemes [2]. To deal with a completely unconstrained syntax, we use a null-grammar network of keywords and fillers. Figure 3 shows this network, which contains HMMs for both keywords and fillers, and allows transitions between all keywords and fillers at any instant in time.

Each keyword in the word spotter is composed of a set of subword HMMs whose form is illustrated in Figure 4. The finite-state Markov chain is characterized by the transition probabilities $a_{i,j}$ for $i, j = 1, \ldots, M$, where M is the number of states (in the figure, M is equal to 3). The particular model shown is known as a left-to-right HMM, which possesses a temporal structure in which lower-numbered states always precede higher-numbered states. The manner in which observations are generated from a state sequence is characterized by multivariate normal observation densities $b_i(\,)$ for $i = 1, \ldots, M$. Speech is represented by cepstrum vector observations that are obtained by using a linear transformation of the short-time log energy speech spectrum [9]. The reader is referred to published tutorials that discuss the maximum-likelihood procedure for estimating the parameters for HMMs of the type shown in Figure 4 [7, 8].

A word spotter is presented with an utterance O and produces the string of words and fillers $\hat{V}$ that results in the maximum *a posteriori* probability, given the input utterance. Thus

$$\hat{V} = \arg\max_V P(V|\mathbf{O}) = \arg\max_V P(\mathbf{O}|V)P(V), \quad (4)$$

where the second equality follows from Bayes rule and because $P(\mathbf{O})$ does not depend on V. In Equation 4, $P(V)$ is the probability that the string of words was uttered. Estimating this probability is a problem in statistical language modeling, which incorporates a variety of information sources including syntax, semantics, and dialog. A considerable successful effort has been devoted to developing language models for many CSR tasks [10], and much of this work may find application in word spotting. For the current HMM word-spotting system shown in Figure 3, however, we assume that all words and fillers are equally probable.

The first term in Equation 4 is the probability that the acoustic utterance is generated for a particular sequence of words. For a single word W in the sequence, this probability is obtained by summing over all possible sequences of hidden states that could have generated the utterance

$$P(\mathbf{O}|W) = \sum_S P(\mathbf{O}, S|W) = \sum_S \prod_{t=1}^{T} a_{s_t, s_{t+1}} b_{s_{t+1}}(O_t),$$

where S is a state sequence of length T. Of course, computing $P(\mathbf{O}|W)$ by exhaustively enumerating all possible state sequences is computationally infeasible because this computation requires on the order of $T \cdot M^T$ operations. Fortunately, this probability can be computed more efficiently by defining the forward probability $\alpha_i(t)$ as the probability of generating all observations up to time t and occupying the HMM state $s_t = q_i$:

$$\alpha_i(t) = P(O_1, \ldots, O_t, s_t = q_i|W).$$

The forward probability at time t can be computed by induction from the forward probability at time $t-1$ as

$$\alpha_i(t) = \sum_{j=1}^{M} \alpha_j(t-1) a_{j,i} b_i(O_t).$$

The full probability of the utterance follows directly as

$$P(\mathbf{O}|W) = P(\mathbf{O}, q_F|W) = \alpha_{q_F}(T),$$

where $s_T = q_F$ is the final state in the utterance. Further discussion concerning the computation of $\alpha_i(t)$ can be found in published tutorial references [7, 8].

In the word spotter, we are interested in finding a

single sequence of words (and fillers) that is optimal, given the observations O in some meaningful sense. The best state sequence through a string of words and fillers can be obtained by using a technique known as the *Viterbi algorithm.* A recursive expression similar to that in Equation 4 can be written for the probability of the best path $v_i(t)$ as

$$v_i(t) = \max_{1 \le j \le M} v_j(t-1) a_{j,i} b_j(O_t). \qquad (5)$$

Equation 5 shows that only a single path (sequence of states) is extended from time $t-1$ to form the optimal path terminating in state q_i at time t.

A trellis structure is used to implement the computation of the Viterbi algorithm. Figure 5 shows a diagram of a simple trellis-structure expansion of a single word in the word-spotting network of Figure 3. This figure illustrates the process of identifying the optimal Viterbi path through a network. The trellis is a time-state representation that displays observation times along the horizontal axis and the HMM state indexes along the vertical axis. For this simple example, the given word model is formed by the concatenation of two subword HMMs of the form shown in Figure 4. For example, the given word model could correspond to the word *go,* which can be expanded as a concatenation of the monophone subword models *G* and *OW.* The small circles represent the hidden states, or nodes, within the subword HMMs, and the large circles represent the grammar nodes shown in Figure 3. The transitions to these grammar nodes are called null transitions because the transition does not produce any output, and therefore does not consume a unit of time. At all nodes, the highest probability path flowing into the node is propagated to the next node, and the most likely sequence of words is recovered by backtracing through a series of pointers that are maintained at the grammar nodes. The likelihood score for a keyword $y_k = L(w_k)$, decoded for observations within an interval of the input utterance, is

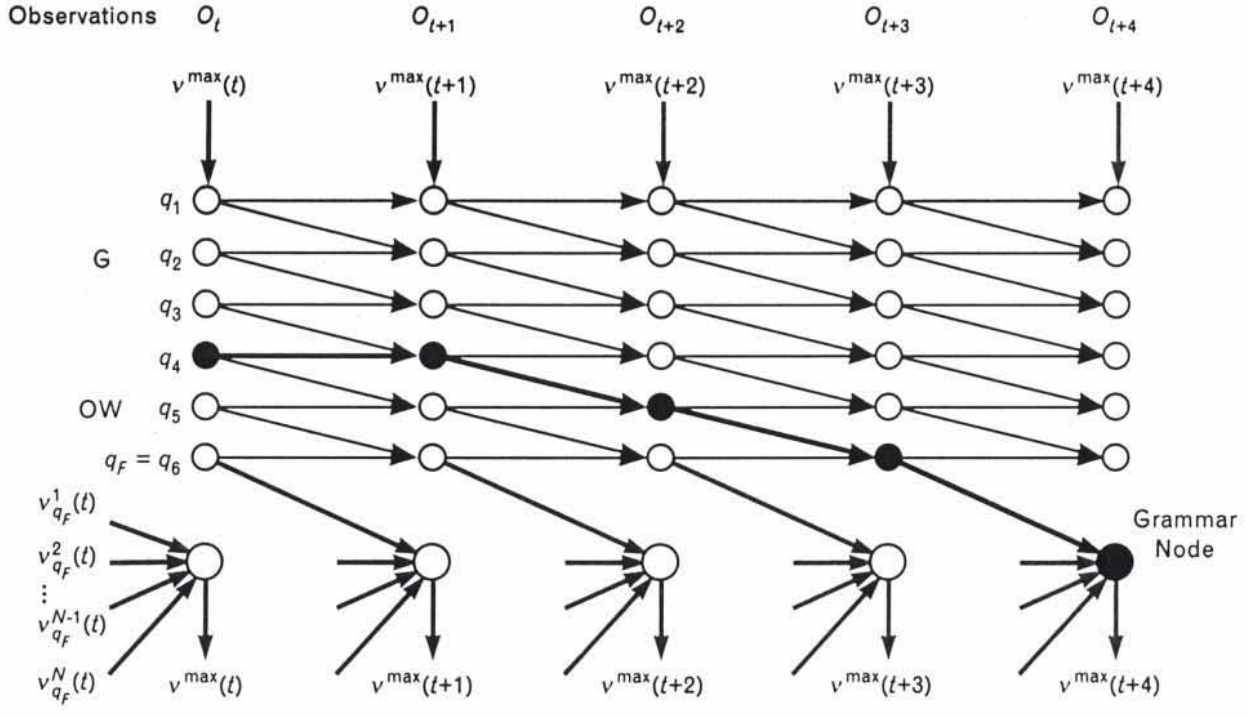

FIGURE 5. Trellis representation of Viterbi search for an observation sequence $O_t, \ldots, O_{t+4}$ through a single word model in Figure 3. The word model representing the word *go* is expanded as two left-to-right subword HMMs of the type shown in Figure 4. Arrows indicate allowed transitions from a source node to a destination node. The small circles represent within-word nodes, and the large circles represent grammar nodes as shown in Figure 3. The optimal path is found by the process of Viterbi search, where only the most probable path is propagated at each node according to the max operation in Equation 5.

WORD SPOTTING FROM A CONTINUOUS UTTERANCE

Software tools have been developed to evaluate the performance of the word spotter. Figure A shows an example of some of the displays produced by these software tools as they appear on the screen of a UNIX workstation. The display at the top of the screen shows the digitally sampled speech waveform for a conversational utterance that is approximately 10 seconds in duration. The vertical lines in this display represent the end points of the putative keyword hits decoded from the utterance by the word spotter, and the labels in the next window correspond to the word names associated with each putative hit. All of the labeled putative hits represent actual occurrences of the keywords in the utterance, except the hit labeled *bingo*, which was decoded when the words *and go* actually appeared in the utterance.

To illustrate the inherent variability in speech processes that makes the word spotting problem so difficult, the portion of the utterance corresponding to the true keyword hit *bangor* is compared to a separate occurrence of *bangor* spoken by a different speaker. An expanded version of the sampled time waveform for this separate occurrence of *bangor* is shown below the original utterance. Two speech spectrograms corresponding to the separate occurrence of *bangor* and the occurrence of the word taken from the original utterance are shown as separate displays at the bottom of Figure A. The spectrograms display the distribution of speech energy in time and frequency, with time displayed along the horizontal axis and frequency displayed along the vertical axis.

Even though the spectrograms show a number of similarities in the two versions of *bangor*, many significant differences also exist. These differences include differences in duration, as well as differences in how the high-energy spectral regions evolve in time. The existence of these natural sources of variability in speech are a fundamental motivation for the use of probabilistic models in speech recognition and word spotting.

passed along to later stages of processing.

The final performance criterion for the integrated speech-message information-retrieval system is the percent-correct classification performance on speech messages. We must also evaluate the performance of the acoustic word-spotting front end, however, because it defines the quality of the partial message transcription provided to the message classifier. An example of keywords located by the word spotter in a continuous utterance is shown in the box, "Word Spotting from a Continuous Utterance." The measure used to describe the HMM word-spotter performance is given as the average probability of keyword detection. The acoustic models were trained by using data collected in a separate data-gathering effort. Keyword sentences were read by 15 male speakers from a 130-word vocabulary, providing an average of approximately 45 occurrences per keyword. The performance of the word spotter was evaluated on 120 speech messages. This corpus amounted to a total of 1.5 hours of speech containing 480 total keyword occurrences. The relative richness of this test set was actually low compared to that of the test set used in another study [2]. This evaluation test set contained a total of approximately 325 keyword occurrences per hour, whereas the conversational speech corpus used in the other study contained the equivalent of 960 keyword occurrences per hour.

The word-spotting performance on the speech messages in the corpus was good at higher false-alarm rates, but poorer than the performance obtained on the test set in Reference 2 at lower false-alarm rates. A 69% probability of keyword detection was obtained at a false-

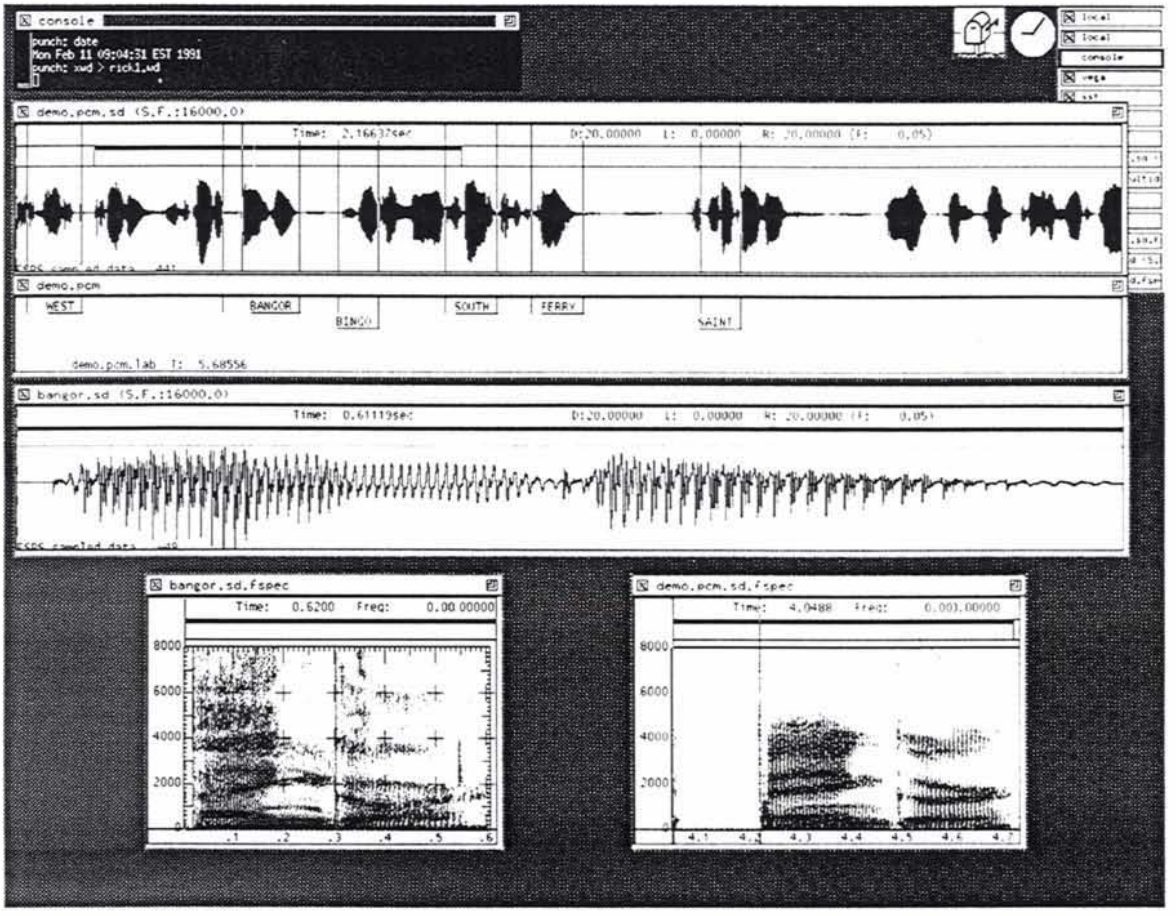

FIGURE A. Displays produced by a set of software tools that were developed to evaluate the performance of the word spotter. The displays are shown as they appear on the screen of a UNIX workstation.

alarm rate of 5.4 false alarms per keyword per hour (fa/kw/hr). The false-alarm rate is given as the total number of false alarms normalized by the number of keywords and the duration of the message. This false-alarm rate corresponds to a total of approximately 330 true hits and 1030 false alarms in the evaluation dataset. A standard figure of merit used in evaluating word-spotter performance is the average probability of detection when averaged over false-alarm rates between 0 and 10 fa/kw/hr. Computing this figure of merit gave 50.2% average probability of detection over 0 to 10 fa/kw/hr, highlighting the poor performance at low false-alarm rates.

Message-Class Corrective Keyword Detection

This section addresses the integration of the maximum-likelihood acoustic word spotter and the mutual-information-based message classifier. The stream of keywords decoded by the word spotter form the partial message transcription that is input to the message classifier. The partial transcription is inaccurate in that it consists of keyword insertions (false alarms) in addition to correctly decoded keywords (true hits). The interest here is in devising a keyword detection mechanism that requires little supervision and can easily be adapted to changing acoustic conditions. A network is described that learns the detection characteristics for all keywords simultaneously through an error metric based on the global message-classification task.

Keyword detection is generally accomplished in word spotting by using a Neyman-Pearson criterion, in which the probability of correct keyword detection is maxi-

mized under an assumed probability of false alarm [11]. The Neyman-Pearson keyword-detection criterion has two primary disadvantages in this context. First, we assume some prior knowledge of the densities associated with the probability of detection and the probability of false alarm for each keyword. This assumption implies significant supervision in training, because these densities are usually estimated from observed likelihoods obtained from speech messages containing hand-labeled occurrences of the keywords. The second and more serious disadvantage is that the adjustment of the Neyman-Pearson operating point is performed individually for each keyword, not in relation to the final message-classification error.

The block diagram in Figure 6 illustrates the model for the speech-message classifier. The word spotter detects keywords in a continuous-speech message and outputs the keyword likelihood scores to the output corresponding to the decoded keyword index. The resulting likelihood scores correspond to putative hits that are either true keyword hits or false alarms; the putative hits, however, are not labeled as true hits or false alarms. If there is more than one detected keyword with the same index, the acoustic likelihood scores for each detected keyword are weighted separately and the average of the weighted scores are presented to the message classifier.

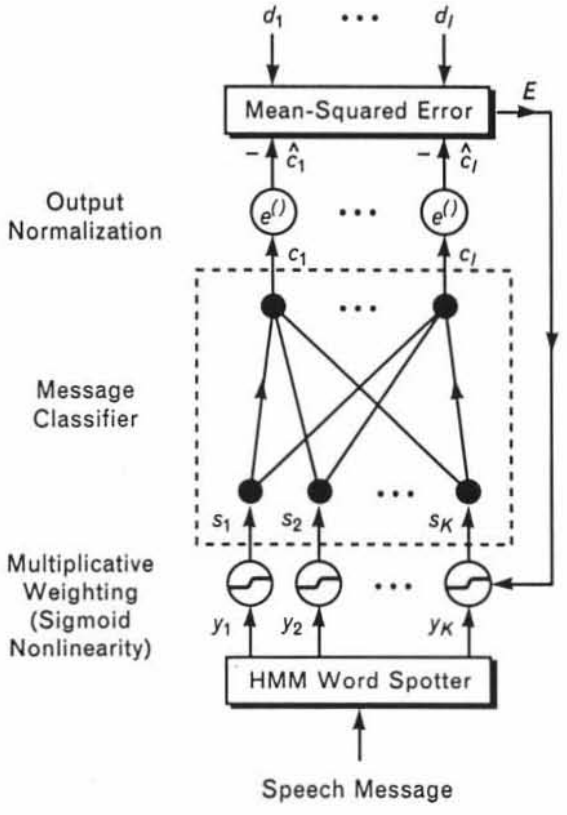

FIGURE 6. Model for a complete speech-message classifier, including multiplicative keyword-likelihood weighting functions. The network learns to interpret keyword likelihoods from the HMM word spotter by minimizing the overall message-classification error.

The input activations to the message-classification network are related to the keyword likelihoods through a set of multiplicative weighting functions. By simultaneously estimating the parameters of these weighting functions, the network learns how to combine keyword scores in a manner that maximizes a criterion relating to the overall message-classification task. A modified mean-squared-error criterion is used to estimate these weights, which implies that the network output c_i represents an estimate of the posterior class probabilities $P(\mathrm{C}_i|\mathrm{M})$ [12]. The weights of the message classifier were estimated by using the procedure described in the subsection entitled "Message-Classifier Model" so that the message-classifier output c_i for class i is an estimate of log $P(\mathrm{C}_i|\mathrm{M})$. An exponential output layer is included to provide the appropriate normalization of the message-classifier outputs. The final network outputs $\hat{c}_1, \ldots, \hat{c}_I$ are normalized so that they sum to 1.

The form of the keyword weighting is a sigmoid nonlinearity

$$s(y_k) = \frac{1}{1 + \exp\left\{-\left(u_{k,1}y_k - u_{k,2}\right)\right\}},$$

where the parameters $u_{k,1}$ and $u_{k,2}$ are estimated by back-propagating the message-classification error

$$E = \frac{1}{2}\sum_{i=1}^{I}\left(d_i - \hat{c}_i\right)^2 \qquad (6)$$

through the message-classifier network. In Equation 6 the quantity d_i corresponds to the desired message-class output for a speech message; the value of d_i is 1 for the correct message class and 0 otherwise. The form for the weighting function can be motivated by observing examples of estimated weighting functions for two keywords. Figure 7 displays the weighting functions for keywords *time* and *west*, and shows the likelihood scores for the

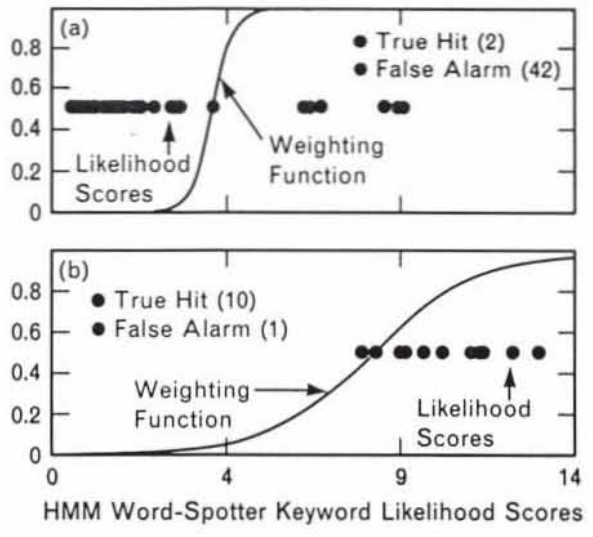

FIGURE 7. Sample keyword-likelihood weighting functions learned through back-propagation of the speech-message classification error: (a) the estimated weighting function for the keyword *time;* (b) the estimated weighting function for the keyword *west.* The likelihoods of putative hits decoded in training speech messages are also shown on each plot.

decoded putative hits superimposed over the plots. For the keyword *time*, shown in Figure 7(a), where most putative hits are false alarms, the estimated weighting function serves to suppress putative hits whose likelihoods in training messages correspond largely to false alarms. Note that this characteristic is obtained as a function of the total message scores, and not as a result of labeled true hits and false alarms. The weighting function for the keyword *west*, shown in Figure 7(b), where most of the putative hits correspond to true keyword occurrences, is different. For this keyword a more uniform weighting is used in the region of all observed putative hits.

The back-propagation equations for estimating the parameters of the weighting functions are easily determined. The message-classifier output is given as

$$c_i = \sum_{k=1}^{N} v_{k,i} s_k(net_k),$$

with

$$net_k = u_{k,1} y_k + u_{k,2},$$

where N is the number of putative hits decoded by the word spotter from the input speech message. The weights $v_{k,i}$, for $k = 1,\ldots, K$ and $i = 1,\ldots, I$, are estimated as described in the earlier section on the message-classifier model. These weights remain fixed during the weighting-parameter update procedure, primarily because of the relatively small number of speech messages. The weight-update equation for the keyword detector parameters is expressed in terms of the message-classification error gradient as

$$u_{k,l}(\tau + 1) = u_{k,l}(\tau) + \eta \frac{\partial E}{\partial u_{k,l}},$$

$$k = 1,\ldots, K, l = 1, 2, \qquad (7)$$

where η is a learning-rate constant. The update interval in Equation 7 corresponds to a single speech message whose duration ranges between 30 and 50 sec. Consequently, the variable τ in Equation 7 is actually a message index and does not represent a fixed time interval. Solving Equation 7 for the message-classification error gradient yields

$$\frac{\partial E}{\partial u_{k,l}} = \sum_{i=0}^{I-1} s_k(1 - s_k) v_{k,i}(d_i - \hat{c}_i) c_i(1 - c_i) y_k.$$

Table 3. Summary of Results for Text Messages and Speech Messages

Text Messages		Speech Messages	
240-Word Vocabulary	GA Search 126 Words	Binary Detection	Corrective Detection
81.3%	78.6%	50.0%	62.4%

Experiments

The performance of the complete end-to-end speech-message information-retrieval system was evaluated on the same 120 speech messages that were used for evaluating word-spotting performance. The putative hits produced by the word-spotter evaluation experiment described earlier were input to the message classifier illustrated in Figure 6. The keyword vocabulary in the complete system evaluation was restricted to a 110-word subset of the total 130-word vocabulary used in word spotting. Table 3 summarizes the results obtained in speech-message information-retrieval experiments, along with results obtained for message classification from test transcriptions of speech messages.

Two separate speech-message information-retrieval experiments were performed. In the first experiment the message-corrective keyword detection was not used, and the message-classifier inputs were activated by the presence of the corresponding putative hit. In the second experiment the parameters of the multiplicative weighting functions in Figure 6 were trained from the 120 speech messages. Each of these messages was labeled by message class. Speech-message classification was then performed by using the weighted keyword-likelihood scores as input to the message classifier. Unfortunately, not enough processed speech messages were available to form independent training and evaluation datasets for evaluating the effect of the message-corrective keyword detection. The performance reported in Table 3 is the speech-message classification performance evaluated on the speech messages used for training the message-class corrective keyword detectors. The third and fourth columns of Table 3 compare speech-message classification performance obtained with and without the message-class corrective keyword detection. For this example the corrective keyword detection resulted in a 25% improvement in performance.

Summary

The most important result of this work is the implementation of the first end-to-end speech-message information-retrieval system. The complete system has been implemented on special-purpose digital signal processing hardware, and demonstrated by using live speech input. The results obtained on a relatively constrained task have demonstrated the feasibility of the technology and also illustrate the need for further work.

Several conclusions can be made as a result of this study. The first conclusion concerns the message-classification performance from near-perfect text transcriptions of speech messages. Even with a perfect acoustic front end, Table 3 shows that a message-classification accuracy of only 78.6% was obtained with a 126-keyword vocabulary. The second conclusion relates to the decrease in performance resulting from the presence of the word spotter. Although test conditions varied somewhat between speech-message and text-message experiments, Table 3 clearly shows that the inclusion of the word-spotting front end results in a significant decrease in performance. Finally, a general comment can be made concerning the effort required to configure a speech-message information-retrieval system to a new task. The most labor-intensive effort in this study was the collection of a separate speech corpus required to train hidden Markov keyword models for the word spotter. This level of effort is clearly unacceptable if the system is to be frequently reconfigured for a new task, as would be the case for the applications suggested at the beginning of this paper. Current research includes the development of techniques to reduce the amount of acoustic speech data necessary for HMM word-spotter training. This effort and other ongoing research are directed toward the development of easily implementable high-performance systems for speech-message information retrieval.

Acknowledgements

We would like to acknowledge the contribution of Marc Zissman for developing tools to evaluate word-spotter and speech-message information-retrieval systems, including the software tools that created the displays in Figure A. Ed Hoffstetter collected training data for the HMM word spotter used in the experiments reported for speech messages. We would also like to acknowledge the assistance of Steve Boll, Alan Higgins, and Dave Vermilyea at ITTDCD in San Diego for providing us with the speech corpus used in our experiments. Comments and suggestions from Gerald O'Leary resulted in significant improvement in the clarity of the manuscript.

REFERENCES

1. A.L. Gorin, S.E. Levinson, L.G. Miller, A.N. Gertner, A. Ljolje, and E.R. Goldman, "On Adaptive Aquisition of Language," *Proc. ICASSP 90, Albuquerque, NM, 3–6 Apr. 1990*, p. 601.
2. R.C. Rose and D.B. Paul, "A Hidden Markov Model Based Keyword Recognition System," *Proc. ICASSP 90, Albuquerque, NM, 3–6 Apr. 1990*, p. 129.
3. R.C. Rose, E.I. Chang, and R.P. Lipmann, "Techniques for Information Retrieval from Voice Messages," *Proc. ICASSP 91, Toronto, 14-17 May 1991*, p. 10.17.1.
4. J.B. Haseler, private communication.
5. A. Nádas, "On Turing's Formula for Word Probabilities," *IEEE Trans. Acoust. Speech Signal Process.* **33**, 1414 (1985).
6. E.I. Chang, R.P. Lippmann, and D.W. Tong, "Using Genetic Algorithms to Select and Create Features for Pattern Classification," *IEEE Proc. Intl. Joint Conf. on Neural Networks, San Diego, CA, June 1990*, p. III-747.
7. A.B. Poritz, "Hidden Markov Models: A Guided Tour," *Proc. ICASSP 88, New York, 11–14 Apr. 1988*, p. 7.
8. L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE* 77, 257 (1989).
9. S.B. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Trans. Acoust. Speech Signal. Proc.* **28**, 357 (1980).
10. F. Jelinek, "Self-Organized Language Modeling for Speech Recognition," in *Readings in Speech Recognition*, eds. A. Waibel and K.F. Lee (Morgan Kaufmann, San Mateo, CA, 1990), pp. 450–506.
11. H.L. Van Trees, *Detection, Estimation, and Modulation Theory* (John Wiley, New York, 1968), pp. 33–34.
12. R.P. Lippmann, "Pattern Classification Using Neural Networks," *IEEE Commun. Mag.* **27**, 47 (Nov. 1989).

RICHARD C. ROSE
is a staff member in the Speech Systems Technology group. He received a B.S. degree and an M.S. degree from the University of Illinois, and a Ph.D. degree from the Georgia Institute of Technology, all in electrical engineering. His areas of research speciality include speech recognition and word spotting, medium-rate speech coding, and speaker identification. Before coming to Lincoln Laboratory in 1988 he was a staff member at AT&T Bell Laboratories. He is a member of Tau Beta Pi, Eta Kappa Nu, Phi Kappa Phi, and the DSP Technical Committee for the IEEE Signal Processing Society.