APPLICATION OF A SEQUENTIAL PATTERN LEARNING SYSTEM TO CONNECTED SPEECH RECOGNITION

A. R. Smith[+], J. N. Denenberg[+], T. B. Slack[+], C. C. Tan[+], and R. E. Wohlford[*]

[+]ITT Advanced Technology Center,
Shelton, Connecticut

[*]ITT Defense Communications Division,
San Diego, California

ABSTRACT

An Experimental Learning Element (ELE) for learning and recognizing sequential patterns is being developed as an adaptable pattern classifier of a larger learning system. Once external patterns are converted into a linear sequence of named objects, the ELE can build models that associate input object sequences with expected output state sequences. The ELE has been successfully demonstrated in learning and recognizing hand-printed characters. This paper describes the ELE and compares its performance with a Dynamic Time Wrap (DTW) based speech recognition system on the task of connected digit recognition. If permitted to continually learn the ELE reaches the same performance level as the DTW-CSR on the same quantized speech test data.

INTRODUCTION

An Experimental Learning Element (ELE) is being developed for learning and recognizing sequential patterns. The ELE task is to learn to associate a sequence of expected output states with a sequence of input states (called objects) so that given a new object sequence, a "reasonable" sequence of output states will be generated. Once external patterns are converted into a linear sequence of named objects, by some task dependent module (e.g., a video capture system or a speech processor front end), the ELE can build models that associate the input object sequences with the expected output state sequences.

Learning Elements will eventually be self-organizing modules of a learning system and should be task independent, adaptable to a changing environment, cellular, and reversible. A learning element is reversible when it can synthesize input sequences when presented with output sequences.

Figure 1 illustrates the analysis or recognition function of the ELE. Output states (X's) are shown to be made up of (or matched to) one or more input objects (Y's). The 'length' of an output state is defined to be the number of input objects it spans. To the ELE there is no inherent meaning associated with an object or an output state, they are simple numbers from finite sets.

As a step in developing the ELE we have tested it on the task of speaker dependent connected digit recognition and compared its performance to that of a dynamic time warp (DTW) connected speech recognition system. In speech recognition the input object to output state sequence relationship can be identified across any pair of quantized levels: centi-second feature vectors to phones, dyads to syllables, demi-syllables to words, and so forth. Although the ELE is applicable between any of the speech levels without change, we choose to use centisecond quantized feature vectors as the input objects and word indices as the output states for easy comparison to an existing DTW speech recognition system. However, there may be better ways of using ELE's to perform speech recognition.

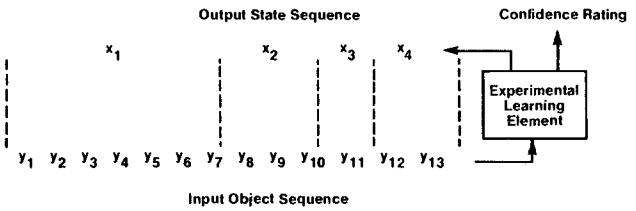

Figure 1: The Analysis Function of ELE

In the following sections we describe the ELE, relate it to other work, and present and discuss the recognition experiments.

SYSTEM DESCRIPTION

Let $y_1, y_2, \ldots, y_T$, or more compactly, y(1:T), be an input sequence of objects to the ELE during time units[1] 1 through T, and x(1:R) be the output sequence of recognized states (see Figure 1). Let b[1:R] be the mapping of input objects to output states such that b_r gives the time unit of the first object for state x_r. The analysis task of the ELE is to find R output states x(1:R) with boundaries in the input sequence of b(1:R) for a given input object sequence y(1:T) such that the probability P(x(1:R), b(1:R) | y(1:T)) is maximized.

The ELE is composed of four parts: modeling, pattern matching, decisions, and learning supervision. An input object enters the pattern matching module where, in the context of preceding objects, it is matched with previously developed models for each output state. This process yields the conditional probability that the input occurs given that the output state occurs. Input based probabilities for each possible output state and length are combined with the probabilities predicted from previously decided output states. The resulting state sequence decision is assigned a confidence rating which is used by the learning supervision module to decide whether or not to update the ELE memory. The following paragraphs briefly describe the four parts.

[1]'Time' refers to the arrival time of the object to the ELE. Whether or not this corresponds to any concept of time in the recognition task depends on the task and the 'front end'.

CH2118-8/85/0000-1201 $1.00 © 1985 IEEE

ELE Modeling

The basic idea behind the ELE model of a sequence is simple. A sequence of objects is learned and modeled by counting the n-grams of objects making up the sequence, where an n-gram is simply a subsequence of n objects. Thus after learning an object sequence for a state, the ELE knows how often each object (1-gram) appeared, how often each pair of objects (2-gram) appeared in any sequence for each state, and so forth up to a specified limit of N. If D is the number of different objects there can be as many as D^N different n-grams. However, the number is limited by the realities of the pattern recognition task. The size of D is determined by the front end process and the number of unique n-grams is determined by the complexity and variability of the states being recognized.

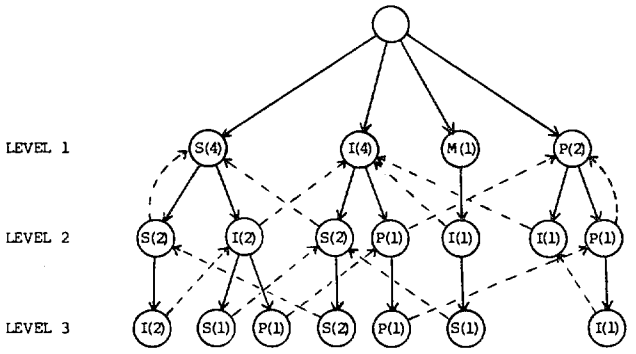

Figure 2: An Example of a Context Organized Memory

The identity and frequency of n-grams is stored in what is called a Context Organized Memory (COM). This memory is a modified tree structure in which each node represents a particular n-gram and is the parent node of all (n+1)-gram nodes that share the same first n objects. In addition, each node is linked to an (n-1)-gram node which represents the same object sequence with one less object at the beginning of the sequence. Figure 2 gives an example in which the object n-grams are composed of letters. The objects on the path to a node at level n define the n-gram represented by the node. The number at a node is the frequency count of the n-gram. The dotted lines show links to the related (n-1)-grams. For example, the 3-gram "SIS" has occurred in the pattern once and is linked to its unique 2-gram "IS". This example was formed from the n-grams ($n<4$) appearing in the word "MISSISSIPPI".

The COM supports an efficient 'Context Driven Search'. The memory arranges the objects so that the set of objects which statistically occur next in context are directly accessible from the current point in the structure. If the next input object does not match any of those in the expected set, the next position searched in the structure corresponds to the less specific context obtained conceptually by ignoring the oldest object and algorithmically by following the link to the (n-1)-gram node. This search technique makes explicit the idea that close context is the best constraint on the identity of the next event. At level n the greatest number of nodes expanded (i.e., searching all sons of a node) before the next object is found will be n. This corresponds to the case when the new object has never been found to follow any subpart of the current n-gram and the search must "drop all context" to look for the object at level 1. An important feature of the Context Driven Search is that the average number of nodes expanded per input object is two. This is obvious if we remember that every failed node expansion (decreasing level by one) is balanced eventually by some successful node expansion (increasing level by one) since the search remains within the finite levels of the tree.

Four types of knowledge are modeled by the ELE in COMs:

Type 1 The frequency of object n-grams forming parts of states;

Type 2 The frequency of n-grams composed of states;

Type 3 The frequency of n-grams composed of state lengths (i.e., the lengths of the underlying object sequence); and

Type 4 The frequency of n-grams composed of items defined by the cross product of states and state lengths.

Knowledge type 1 relates object sequences to states. An object n-gram which appears in more than one state is stored once and the node lists the proper states with frequency counts. In addition, the frequency count for each state listed in the node is further broken down into frequency counts for each occurring position within the state. This is a detailed position given by the number of objects preceding and the number of objects following the object n-gram. Currently, however, the pattern matching algorithm generalizes the position information to determine how often the n-gram occurred in each one-third of a state. More detail is stored now than is used.

The COM structures for the remaining knowledge types are less complex, similar to the example of Figure 2. The information in these COM's are used to compute the Predict Probability described in the next section.

ELE Pattern Matching

Consider an object 4-gram, $y_1y_2y_3y_4$, stored at node j and let f_j be the frequency of occurrence of the 4-gram and f_i be the frequency of occurrence for its parent node, a 3-gram. Then the conditional probability that object y_4 occurs in the context of the 3-gram is given by the maximum likelyhood estimate:

$$P(y_4 \mid y_1y_2y_3) = f_j/f_i. \quad (1)$$

This is the probabilistic basis for pattern matching in the ELE.

Using conditional probabilities retrieved from COMs the ELE computes at each time interval, t, two basic probabilities:

1. Input Probability: the probability that the input object sequence beginning at time b occurs and spans a state given that it ends at time t and the state occurs. This is based on knowledge type 1. This part of the algorithm would be called the template matcher in a speech recognition algorithm.

2. Predict Probability: the probability that a state and length occurs given that a previous sequence of states and lengths have occurred. This is based on knowledge types 3, 4, and 5. This process can be related to a 'syntax control' module of a connected speech recognition algorithm.

ELE Decision Process

From the input and predict probabilities at each input time, t, the Decision Process computes the probability that a state and a length and the input object sequence spanning that length ending at t occurs given past context. These probabilities are combined over time using the Viterbi algorithm [1] to compute the k most likely state sequences ending at time t, for some k. The most likely state sequence ending at final time T is the recognized state sequence.

ELE Learning Supervision

The Learning Supervision module decides whether or not to learn to associate the output state sequence with the input object sequence. Learning the next object in a sequence is simply a matter of creating a new node in the tree whenever the object appears in a new context or incrementing a frequency count when it appears in a previously learned context. The object is learned in all possible contexts from the (n-1) gram preceding it for some maximum n down to a null context in which the object is recorded by itself as a 1-gram. An object can also be 'unlearned'. This is identical to learning except that node occurrences are decremented and nodes are deleted if their occurrences become zero.

The decision to learn is based on a threshold test of the confidence factor and external reinforcement. External reinforcement may be either from another ELE or from a human operator. The reinforcement may also include corrections to some of the state and boundary decision made by the ELE. These corrections are passed on to the ELE Modeling section before the COM structures are updated.

RELATED WORK

The model used by the ELE is that sequences are probabilistic functions of Markov processes. We are using a variable order Markov process where for each Markov **state**[2] the order is equal to one minus the level of the node in the corresponding COM tree. Roucos, Makhoul, and Schwartz [2] recently used a tree structure to represent a variable-order Markov chain for modeling the output of a variable frame rate LPC vocoder.

An nth order Markov process is equivalent to some first order Markov process with an expanded state space. In fact, the ELE learning process maintains such a **state** expansion automatically. Each node on a COM tree represents a **state** of the Markov chain encoding a particular n-gram. The transitions to all possible next Markov **states** are given by the links to all sons of the node. New Markov **states** are added as new n-grams are observed and can be deleted as transitions to them become relatively improbable.

Markov **states** in the ELE are directly related to the input of the recognition task in contrast to the **states** in the frequently used hidden Markov model [3, 4, 5, 6]. A hidden Markov **state** stochastically models both the number and identity of variables in a random variable sequence with the potential of loosing important detail. Also, hidden Markov model generation is computationally expensive and non-incremental whereas learning in the ELE is simple and incremental. The hidden Markov model has the advantage that memory requirements are fixed whereas pruning of unlikely nodes in the ELE must be done at some stage to control memory growth.

EXPERIMENTS AND RESULTS

The ELE was tested on random connected digit phrases recorded by five male and five female speakers. Each speaker produced 50 three digit and 40 five digit phrases. Eight of the three digit and thirteen of the five digit phrases were set aside as training phrases leaving 69 test phrases containing 261 digits per speaker. Three isolated samples of each digit were recorded from each speaker and were added to the six digit samples extracted from training phrases to give a total of nine templates per digit per speaker. This training and testing data was again produced in a second recording session by each speaker several days after the first recording session. The procedures differed between the two sessions only in that the training samples were extracted by hand from the first session data and by an automatic procedure from the second session data.

The speech signal was processed by a 16-bandpass filter bank to produce filter coefficients every 10ms. Each vector was linearly transformed from 16 filter coefficients to 10 mel-frequency cepstrum coefficients [7]. The data was then time compressed by a variable frame rate encoding technique which eliminated on the average one-half of the frames. A set of about 130 most representative coefficient vectors was determined from the training data for each speaker and used for vector quantization [8]. Thus, the input domain for the ELE was defined by about 130 different objects. The output domain of the ELE was defined by the 10 digit vocabulary and a silence state.

Since the digit sequences were known to be random, the ELE was prevented from predicting any state (i.e, digit), length, or state-length based on previously identified sequences of states, lengths, or state-length pairs. Therefore, in these experiments, only the models built up in knowledge type 1 were used in the testing. The COM tree structure was limited to a depth of 5 so that n-grams longer than 5 objects were not learned.

Performance results were compared to a DTW-CSR system similar to one described by Bridle, Brown, and Chamberlain [9]. The CSR system has the ability to run on non-quantized data, one-sided quantized data (only the template data vector quantized), and two-sided quantized data (both template and test data quantized before matching). The CSR system was run in all three modes to indicate the limitations the front end processing was placing on the ELE performance. The ELE at this time requires two-sided quantization although it is possible to extend it to handle one-sided quantization.

Both systems were trained for each speaker on the same data. The CSR system formed a template from each digit sample to give a total of 90 templates plus a silence template. The ELE built its COM tree structure from the same template data. When the CSR system was tested in the two-sided quantization mode, the input to both systems was identically for each test phrase. The word recognition rates are given in Table 1.

Training Session	1	2	1	2	Column
Testing Session	1	1	2	2	Average
Non-quantized CSR	99.4	98.1	97.3	97.6	98.1
One-sided CSR	99.4	98.0	97.4	97.5	98.1
Two-sided CSR	98.7	97.5	96.1	97.1	97.4
Two-sided ELE	97.0	90.7	92.2	95.0	93.7

Table 1: Performance of ELE and CSR on Connected Digits

A second experiment tested how ELE performance changed when learning while testing (LWT) took place. Memory was initialized with the same training used above. The test phrases of each speaker were ordered to alternate between three digit and five digit phrases. After every phrase, the ELE compared its recognized digit sequence to the correct sequence and

[2] We will use 'state' in bold type to differentiate a Markov **state** from an output state of the ELE.

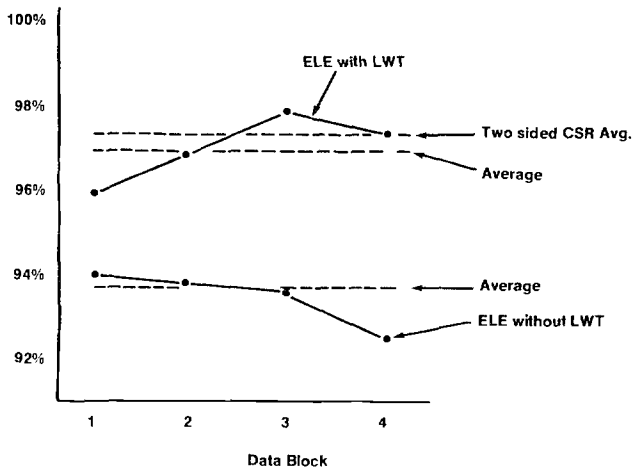

Figure 3: Performance with and without LWT

determined what digits were correct, substituted, inserted, or deleted. Every digit that was correct and was adjacent on both sides to a correct digit, a recognized silence, or a phrase end was learned by updating the state model with the underlying object sequence. The constraints on adjacent recognitions help to assure that the end points of the digit are correct. Similarly every digit that was inserted or substituted in place of the correct digit and was adjacent on both sides to a correct digit, a recognized silence, or a phrase end was 'unlearned'.

The effect of more and more learning can been seen by dividing the data into four blocks each containing approximately 650 digits over all speakers. Performance statistics are then collected from each block. The graph of Figure 3 combines the results over the four test and training session pairs.

DISCUSSION

The CSR system results in Table 1 show that vector quantization on the test data (but not on the training data) accounts for about 16% of the performance difference between the non-quantized CSR and the ELE system (from the last column we have (98.1-97.4)/(98.1-93.7)). Overall the ELE has about 2.4 times the errors that the two-sided CSR system has with the same amount of training.

What limitation has been placed on the ELE that is important for speech recognition? One is immediately obvious. The ELE has no knowledge of any similarity metric for the input object domain. There is no concept of a frame of speech in the test data being similar to a frame in the template data. An object (quantized frame) either occurs or does not occur in some previous object context for a state. If it does not occur the match process goes up the levels in the COM structure (i.e., dropping old context) until the object is found with a correspondingly lower conditional probability or until it is determined to not exist even at level 1 at which point a lower limit default conditional probability is used. Without the concept of object closeness the ELE has a weak model of speech which requires more training to obtain good performance.

The graph shown in Figure 3 supports this. Even within the first block of data the ELE has removed one-third of its errors by learning from its successes and errors. This continues in the second block and in the last two blocks two-thirds of the errors have been corrected and the performance is greater or equal to the average two-sided CSR performance at 97.4%. It is not known what the individual block performance is for the CSR but the ELE performance without LWT suggests that it would drop for the last block. It is also not known how the performance of the CSR system would change if it also added templates to its data base. However, these templates could not be simply added since the throughput of the CSR system is linear with the number of templates it uses. Although, 230 digit samples were added to the original 91 samples in the ELE data base per speaker during each LWT experiment, the processing time was only 25% greater. This processing time included the time to update the COM structures.

The performance results were similar for the four pairs of training and testing sessions except for training and testing within session one. In that experiment, the CSR performance of 98.7 was not obtained in data blocks three or four (the ELE had 98.0 and 97.8, respectively). This may suggest that for the current ELE techniques there is an upper limit to speech recognition performance independent of training when one ELE is used to bridge between quantized speech and words. We are continuing to investigate the ELE and intend to remove any limitations while retaining the systems basic view of sequential patterns.

References

1. G. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE.*, Vol. 61, March 1973, pp. 268-278.

2. S. Roucos, J. Makhoul, R. Schwartz, "A Variable-Order Markov Chain for Coding of Speech Spectra," *Proc. ICASSP.*, 1982, pp. 582-585.

3. J. K. Baker, "Stochastic Modeling for Automatic Speech Understanding," in *Speech Recognition*, R. Reddy, ed., Academic Press, 1975.

4. S. Levinson, L. Rabiner, and M. Sondhi, "Speaker Independent Isolated Digit Recognition Using Hidden Markov Models," *Proc. ICASSP.*, 1983, pp. 1049-1052.

5. P. Brown, C. Lee, and J. Spohrer, "Bayesian Adaptation in Speech Recognition," *Proc. ICASSP.*, 1983, pp. 761-764.

6. R. Billi, "Vector Quantization and Markov Source Models Applied to Speech Recognition," *Proc. ICASSP.*, 1982, pp. 574-577.

7. S. Davis, P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Trans. on ASSP*, August 1980, pp. 357-366.

8. B. Landell, J. Naylor, and R. Wohlford, "Effect of Vector Quantization on a Continuous Speech Recognition System," *Proc. ICASSP.*, 1984, pp. 26.11.1-4.

9. J. Bridle, M. Brown, R. Chamberlain, "An Algorithm for Connected Word Recognition," *Proc. ICASSP.*, 1982, pp. 899-902.