SPOKEN LANGUAGE UNDERSTANDING FOR PERSONAL COMPUTERS

George M. White
David Nagel

Apple Computer Inc
20525 Mariani Avenue
Cupertino, CA 95014

ABSTRACT

Automatic speech recognition technology will soon allow users to converse with their computers. This paper describes an approach to improve the human interface through speech recognition; describes how our research benefits the DARPA Spoken Language Research program; and describes some research results in the area of merging Hidden Markov Models (HMM), and Artificial Neural Nets (ANN).

We apply ANN techniques as a post-process to HMM recognizers such as Sphinx. We show that treating ANNs as a post-process to partially recognized speech from an HMM pre-processor is superior to using ANNs alone or to hybrid systems applying ANNs before HMM processing. A theory explaining the advantages of applying ANNs as a post-process is presented along with preliminary results.

IMPROVING THE HUMAN INTERFACE WITH SPEECH RECOGNITION

Apple's approach is distinguished by its emphasis on conversational communication with personal computers as distinct from dictation or command and control only. It is further distinguished by integration of speech recognition into the visual "desk top" metaphor of personal computers. We believe that speech recognition will impact personal computing sooner and more effectively if it is integrated with other I/O modalities such as the mouse, keyboard, visual icons, dialog boxes and perhaps speech output. We expect to bring such integrated systems to market in the 1990's.

Our approach is similar in spirit to notions of Alan Sears in his SLIP (speech, language icons, and pointing) paradigm but with some distinctive differences. We will use task domain constraints provided by particular application packages on personal computers to create constrained natural language understanding. Furthermore we will implement interactive voice and text response mechanisms such as dialog boxes and speech synthesis to respond to the users input. We will provide a conversational natural language understanding within narrow task domains on personal computers in which speech is augmented with pointing, typing, and mousing around.

SPEECH UNDERSTANDING ON PERSONAL COMPUTERS

A perennial problem confronting the speech recognition community has been lack of adequate computing power to perform real time recognition and understanding. This shortcoming is being solved, not so much to serve speech interests as it is to serve the computing needs of society at large. It is the natural progression of VLSI, economies of scale of mass produced personal computers, and computing infrastructures.

For personal computer users, speech recognition is particularly useful in areas where the user is confronted with too many options to easily manage with function keys or a small number of shift-key combinations. The current solution is to use pull down or pop up menus but these are fast becoming less convenient by shear weight of numbers of options. Sub-directories of sub-directories are becoming common. The arm motion simply to get the initial menu, and then each submenu, is a limitation on ease-of-use. Speech recognition can cut through the branches of the menu tree to speed throughput as

long as the speech recognition is fast and accurate enough.

Speech recognition offers other advantages to the user interface by allowing many words and phrases to mean the same thing. If a user forgets a command or does not know if one exists, speech recognition systems can partially solve this problem by supporting synonyms and paraphrase. In addition, when user defined scripts and macros become numerous, they are difficult to manage with function keys and shift key commands. Speech recognition allows users to invoke these macros and scripts with a distinctive name or phrase and avoids function keys altogether.

We expect to employ speech in interfaces to educational programs, standard computer applications (spreadsheet, word processing, etc.), multimedia systems, and telephone access systems.

Automated language learning is another area of particular interest to Apple that seems to be yielding to DARPA sponsored research. Speech recognition techniques are becoming good enough to time align known utterances to templates for the words in the speech. Words that are poorly pronounced can be spotted and students can be directed to repeat offending words to mimic correctly pronounced words from the computer.

COMMERCIAL APPLICATIONS OF DARPA TECHNOLOGY

Our philosophy at Apple is to leverage the efforts of other companies and researchers by providing them a platform through which they can commercially address the needs of personal computer users. For example, we stay out of certain business areas such as selling application software in order to encourage independent developers to develop products in these areas. In the research area, we stand ready to adopt systems developed by DARPA contractors and offer them along side our internally developed systems to commercial outlets.

Apple encourages outside vendors to produce ASR systems to be promoted or sold by Apple. We prefer to work with those DARPA contractors that make their research freely available, but we will also consider licensing technology from the outside if it is better than internally developed technology. We actively seek partners to supply components to be used in our own internal ASR systems.

For example, we currently have SPHINX working on a MAC which we call MACSPHINX. This is not currently scheduled to be shipped as a product, but a product may be based on MACSPHINX at a later time.

As our contribution to the underlying technology, we intend to extend Sphinx to give it a speaker dependent mode in which it can learn new words "on the fly". We will initially do this by augmenting Sphinx with ANNs as described below.

As another example of partnering, we expect to begin building on Victor Zue's work with VOYAGER. We will receive VOYAGER from MIT in a month or two running on a MAC platform. We expect to modify it to run faster and with an order of magnitude less computing power.

THE PLUS SPEECH ACCELERATOR PROJECT:
In order to make it easier for DARPA contractors to use MACINTOSH computers, and to build speech recognition systems that would control applications on MACINTOSHs, we have supported and encouraged Roberto Bisiani to design a "speech accelerator" for more than a year. The goal was to allow a MAC to have intimate control over an accelerator processing unit that would offer between 50 and 200 MIPS economically and with broad base of software support. This was achieved in an external box, named PLUS by its designer Roberto Bisiani, which has a SCSI interface as well as higher speed NU BUS connection to a MAC. The SCSI interface allows the box to be programmed by other computers such as SUN computers as well as using a MAC. However, the high speed NU BUS interface to the MAC will allow tighter integration with the MAC than other computers. The box itself contains Motorola 88000s, one to ten in a single box; and the boxes may be daisy chained. We hope many of the DARPA contractors in attendance here will use the accelerator box to make their spoken language communication systems available to MAC

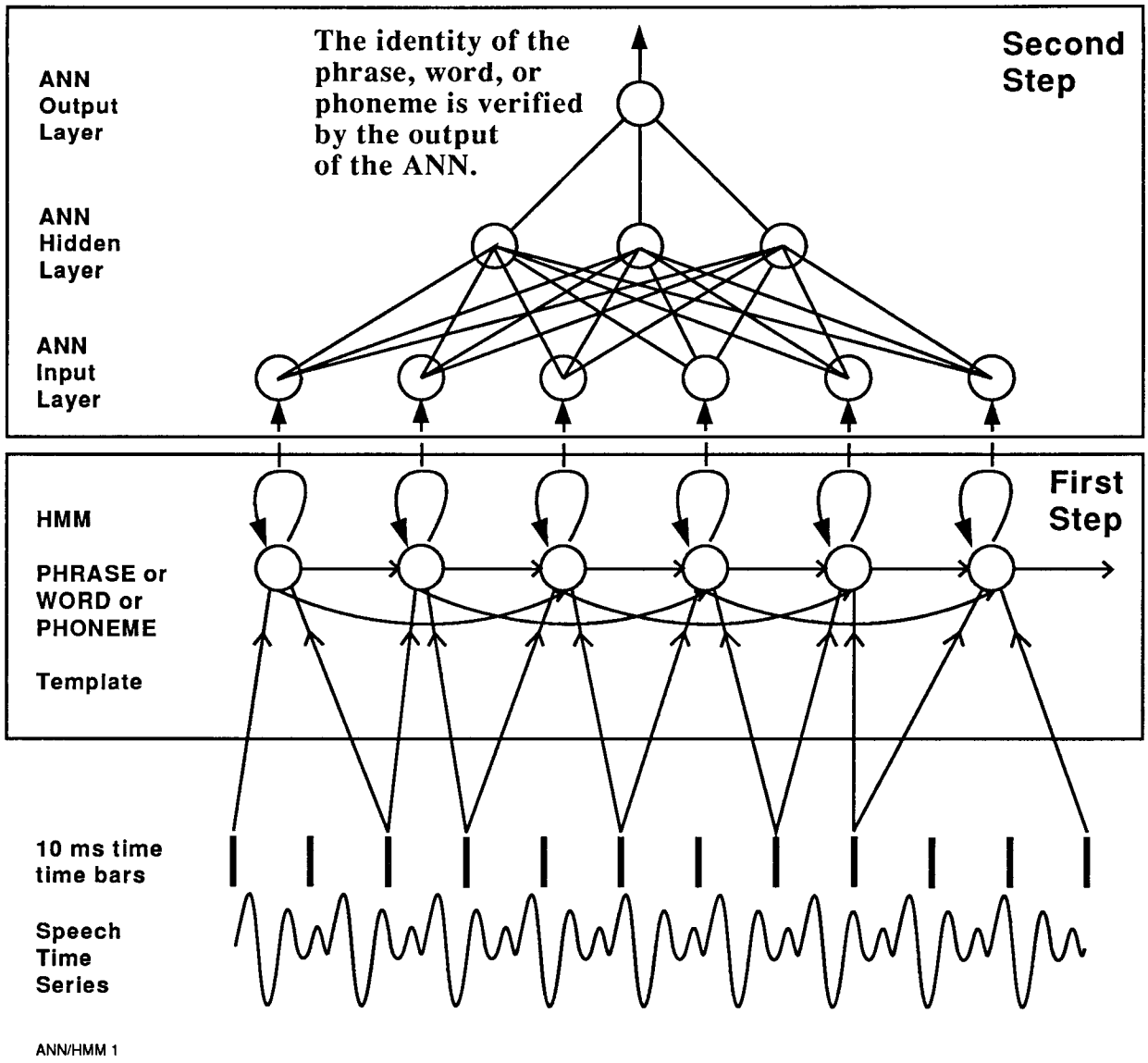

FIG. 1 Nodes in a canonical HMM topology pointing to time intervals in speech time series and also to the input nodes in an ANN. The pointers to the time intervals are established using well known techniques as part of the HMM processing in step 1. Standard ANN techniques are then applied in step 2 to the speech which has now been time aligned to fixed structure of the HMM.

applications. Development of this box is currently funded by DARPA and will probably be available later this year.

ANN POST PROCESS TO HMM

The Hidden Markov Model (HMM) approach is the dominant speech recognition paradigm in the ASR field today. Millions of dollars have been spent in dozens of institutions to explore the contributions of HMM techniques to speech recognition. Artificial Neural Net (ANN) technology is newer, but it has also become heavily funded and widely investigated. It has been only within the last year or two that the possibility and need to combine techniques from these two fields has emerged. It is very likely that numerous proposals for merging HMMs and ANNs will be presented in the next few years.

George White has proposed a new and previously unpublished, technique for combining HMMs and ANNs. We refer to this technique as the ANN "postprocessing technique" by which we mean that

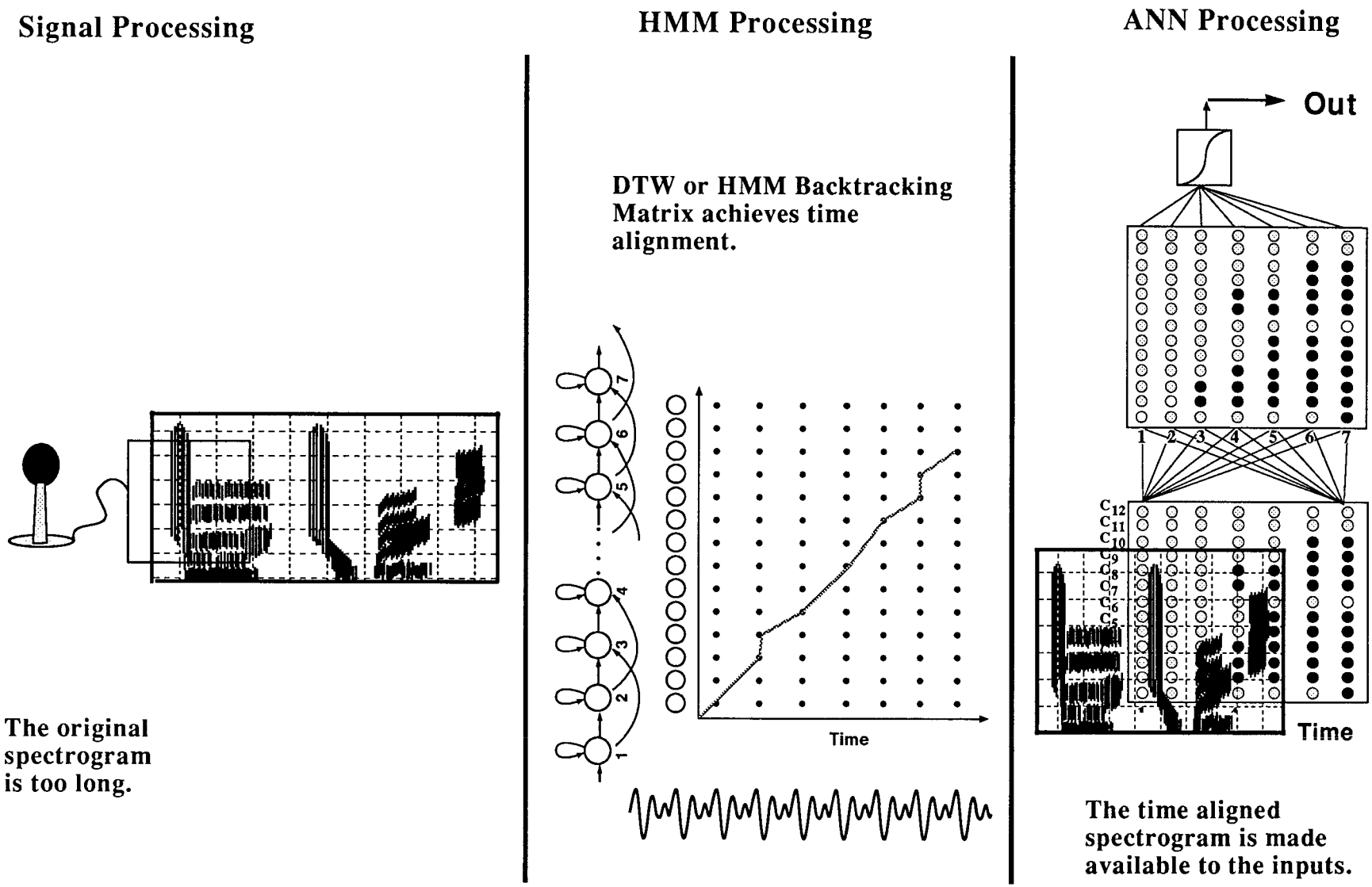

Fig. 3: A third view of the process shown in Fig 1 and 2. This view emphasizes the time alignment achieved by HMM prior to ANN processing. It explicitly shows the back tracking matrix used to achieve time alignment.

ANNs should be applied to speech "after" HMMs have been applied. The HMM processing determines where in time words should be located in the acoustic input data. More to the point, input nodes in an ANN structure may be connected to states inside the finite state machines that form HMMs, which are then connected to time intervals in the unknown speech. The HMMs accomplish non-linear time warping, otherwise known as "dynamic time warping," of time domain acoustic information to properly match the rigid structure of a neural net template (see Fig 1). We postulate that there is great significance to bringing inputs from HMM states that span several time units to provide input to the neural nets.

The fundamental postulate of HMM or DTW (Dynamic Time Warping) is that the speech sound similarity scores in adjacent time intervals may be simply summed up provide a global match score. This is rationalized by assuming that the probability of global match over all time intervals, P(t1,t2,t3,...tn), is equal to the product of the probabilities of the matches for each individual time interval.

In other words, **the fundamental assumption behind HMM or DTW is:**

P(t1,t2,t3,...tn)=P(t1)P(t2)P(t3)...P(tn) Eq.1

This may be acceptable when there is no practical alternative but it is not accurate and can lead to recognition errors when when subtle differences between words matter.

ANNs can circumvent this problem if they are trained on the global unit spanning t1,t2,t3,...tn. The fundamental motivation behind our approach to merging ANNs and HMMs is that

ANNs compute P(t1,t2,t3,...tn) directly

and thus avoid the error of Eq 1.

For example, the HMM approach to scoring word sized units sums scores for phonemes which in turn sum scores over elemental time units, typically 10 ms in duration, which assumes statistical independence between the phonemes and also between the 10 ms domain units. Since these units are usually not statistically independent, some are typically over-weighted. ANNs spanning word sized units overcome some of these limitations.

Previous work on the general subject of merging HMMs and ANNs includes, "Speaker-Independent Word Recognition Using Dynamic Programming Neural Networks", by Sakoe, Isotani, and Yoshida (Readings in Speech Recognition, edited by Alex Waibel & Kai-Fu Lee). Other work includes "Merging Multilayer Perceptrons and Hidden Markov Models: Some Experiments in Continuous Speech Recognition" (ICSI, TR-89-033, July 1989), which evidently applies MLP (a form of ANN) to individual states inside HMM models. While this is a merger of ANN and HMM techniques, it falls short of the power of an ANN post process which overcomes the lack of statistical independence between adjacent time intervals.

Other work includes "Speaker-Independent Recognition of Connected Utterances Using Recurrent and Non-recurrent Neural Networks" (IJCNN, June 1989). This work, like the one mentioned above, doesn't propose to achieve time alignment by HMM techniques as a precursor to applications of ANNs which is the basis of our proposal. Instead, it proposes to apply ANN technology first and then apply HMM techniques. This necessarily precludes the beneficial effects of HMM guided dynamic time warping from being realized by the inputs to the ANNs.

Other related work in this area may be considered as special cases of one of the two above mentioned approaches.

GENERAL COMMENTARY ON ANN

While we advocate the use of ANN in conjunction with HMM or DTW, we do not at all endorse the notion that ANNs should be used alone, without DTW or HMM or other segment spotting approaches. Internal time variability in word pronunciation in multiple pronunciations of the same word must be managed and ANNs have no easy way to handle temporal variability without extraordinary requirements for silicon area.

To handle the time variability problem with

accuracies competitive with HM, neural net structures must store intermediate results inside the neral net structures for each time interval. For problems as complex as speech recognition, this is not practical on silicon because of the number of interconnections is limited by the two dimensional nature of the surfaces of silicon chips. Trying to simulate the needed structures on Von Neumann machines, with DSPs for example, will result in optimal solutions similar to the Viterbi search currently used in HMM systems. In other words, as long as we are restricted to two dimensional chips and Von Neumann architectures, ANN simulations will necessarily need to employ search strategies that are already core technologies in the speech community. It would be misguided for the ANN community to ignore these refined search techniques. It is not likely that the need search strategies can be circumvented as long as the dynamic allocation of cpu operations is needed and it will be needed until we achieve three dimensional interconnections between ANNs. We should expect that hybrid combinations of HMMs (or DTW based approaches) and ANNs will be superior to pure ANN systems until an entirely new process for producing three dimensional integrated circuits is invented, and this will probably be a long time.

RESULTS

These ideas have been adapted by Parfitt at Apple for application to Sphinx. Parfitt modified the Sphinx recognition system to generate input to a three layer perception, a type of ANN as shown in Figures 1 and 2. The following describes his implementation: The Sphinx system is initially trained in the traditional manner using the forward/backward algorithm. However, during training and recognition, a modified Viterbi/beam search is used. A record with backpointers is maintained for all nodes that are visited during the search. When the last speech interval is processed, the best final node identifies the optimum path through the utterance. The mapping of the speech data to the optimum path is used to establish word boundaries and derive a set of time aligned paramaters to pass to the ANN.

A separate ANN is used for each word in the vocabulary. Each word in the vocabulary is represented by one or more triphone models. Although each triphone has seven states, only three are unique states. Because the HMM models contain skip arcs, the speech can skip one, two or all three of the states. The model also contains self-loop arcs for each of the states. The speech may match a given state an arbitrary number of times. When several speech samples match a given state, the middle sample is used to supply input to the ANN. When an even number of samples match, the left middle sample is used. When no speech samples match a given state, zero is used as input to the ANN.

The ANN uses different input parameters than the HMM triphone in SPHINX. The SPHINX recognizer works on three VQ symbols per window. The windows are twenty milliseconds wide and are advanced by ten millisecond increments. The VQs are derived from three sets of parameters, twelve Cepstral Coefficients, twelve Delta Cepstral Coefficients, and Power/Delta-Power. However, the ANNs do not receive VQs. Instead, they receive the same three sets of parameters before they are vector quantized except that each parameter is linearly scaled to range between minus one and plus one.

As shown in Figure1, the ANN models have one hidden layer and a single output node. Words are constructed from a sequence of triphones. For example, for the word "zero", there are four input triphones. Each triphone has three unique HMM nodes and each node has twenty-six input parameters for a total of 78 inputs per triphone. Hence, the word "zero" has 312 inputs to the ANN. The 78 inputs per phone are fully interconnected to 25 nodes in the hidden layer. All the hidden layer nodes are fully interconnected to the single output node.

ANN TRAINING: Each word ANN is trained from time aligned data produced by the modified SPHINX. Two sets of data are used to train the ANNs. One set of data represents the "in class" utterances. The ANN is trained to be plus one when this first set of data is presented. The second set of data represents "out of class" utterances and is composed of other words. For this case, the output of the ANN is trained to be minus one. The ratio of "out of class" utterances to "in class" utterances is 3.5. The ANNs tend to converge to 100% accuracy on the training data after about 300 passes through the data. Back propagation is used for training.

ANN RECOGNITION: The time normalized data for each word from the utterance is fed as input into each of the neural nets. If the best path word is shorter than a given neural nets' input, additional data is taken from the rest of the best path. Silence is always skipped. If the end of the utterance is reached before enough data is collected, nulls are input to the neural net. For recognition, the individual neural nets are connected together and the output which is most "on" is used to indicate what word.

The system was tested on TI Connected Digits Database. Six male speakers from two different dialects were used for training. Three males, MKR, MRD, and MIN were taken from the Little Rock, AR dialect. The other three male speakers, MBN, MBH, MIB, were taken from the Rochester, NY, dialect.

The current modifications to Sphinx only produce pointers to the best candidate words during recognition. There are three classes of errors: insertions, deletions, and substitutions. When the HMM scores correctly, the ANN was tested and is in agreement 100% of the time. For the three classes of errors, only substitution errors have been tested with the ANN. From a set of 385 utterances, the Rochester male speakers, nine substitution errors were made by Sphinx. The ANNs corrected four of the nine errors.

CONCLUSIONS: A larger set of data needs to be tested before any strong conclusions can be drawn. The initial reduction in error rate by 44% of an already highly tuned system is promising.

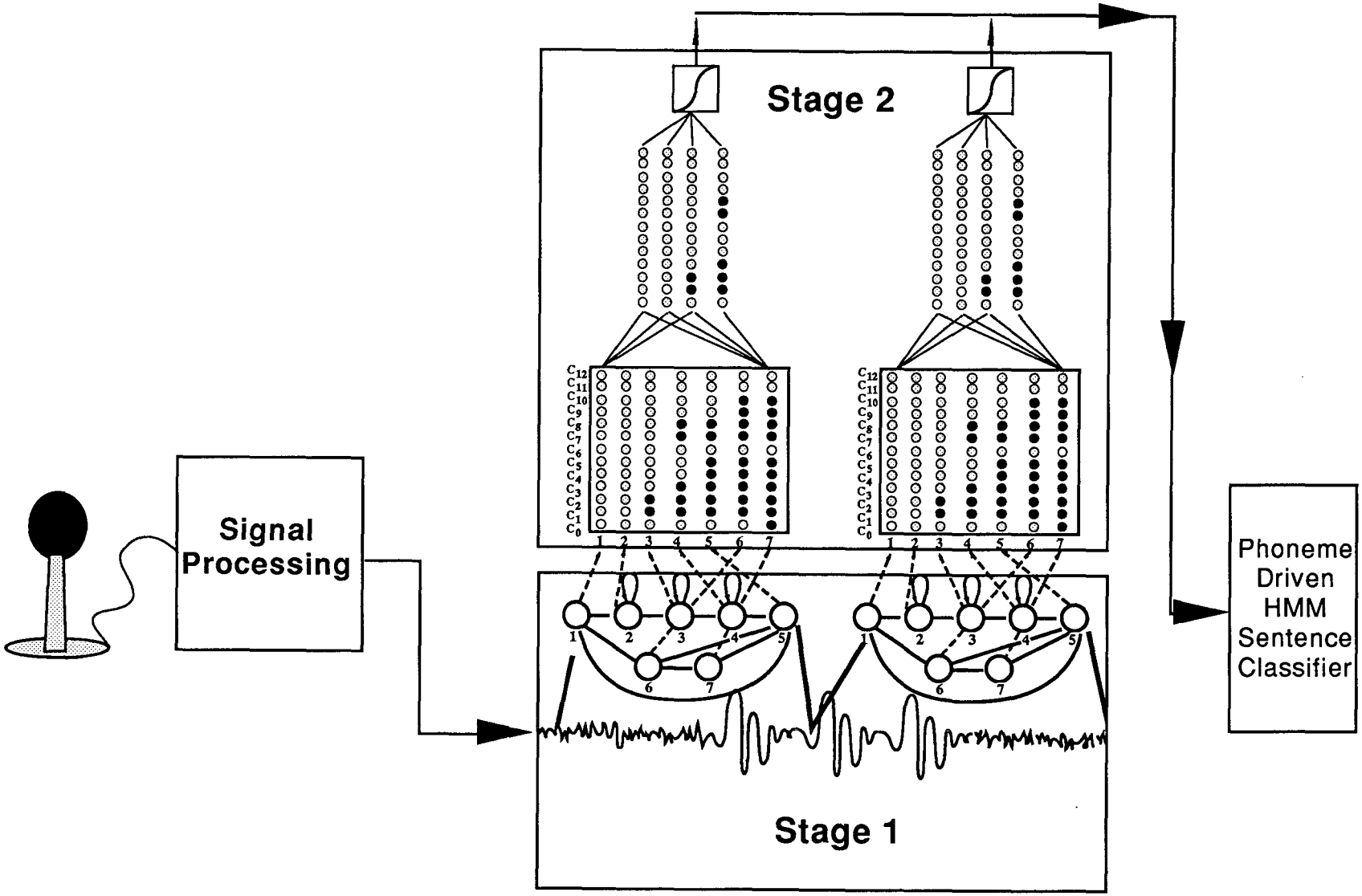

Fig. 2: Different view of the same operations as shown in Fig 1. This view emphasizes the treatment of phonemes and triphones as embodied in the Sphinx recognition system of CMU. Speech input is segmented by the HMMs in stage 1, and the HMM triphone models feed the ANNs in stage 2.