DEVELOPMENT of SCHEMATA DURING EVENT PARSING:

Neisser's Perceptual Cycle as a Recurrent Connectionist Network

Catherine Hanson
Department of Psychology
Temple University
Phildelphia, PA 19122

Stephen José Hanson †
Learning Systems Department
SIEMENS Research
Princeton, NJ 08540

Abstract

The present work combines both process level descriptions and learned knowledge structures in a simple recurrent connectionist network to model human parsing judgements of two videotaped event sequences. The network accomodates the complex event boundary judgement time-series and provides insight into the activation and development of schemata and their role during encoding.

Perceiving and Encoding Events

Day to day experience is characterized, remembered, and communicated as a series of events. We think about *driving to work*, we remember *having an argument* with our spouse, and we tell a friend about our plans to *attend the theatre* next Saturday. Abreviated phrases such as *driving to work* act as a type of shorthand notation for describing complex action sequences. Thus, our abilty to communicate sucessfully with others using such labels as *driving to work* reflects a certain level of familiarity with the referenced activities that we share or presume to share with our intended audience.

How common is our knowledge about common events? Empirical work suggests that there is considerable consensus concerning the constituent actions of familiar events (Bower, Black, & Turner, 1979). Bower, et al. found that subjects showed considerable agreement about the composition of common events (e.g., *going to a restaurant*), many responses being offered by more than 70% of their subjects and very few being unique. Considerable agreement about event boundaries extends to online measures of parsing as well (e.g., Newtson, 1973; Hanson & Hirst, 1989) suggesting that familiarity with events may provide the basis for understanding and encoding new information.

Neisser's Perceptual Cycle

Neisser (1976) has suggested that perception is a cyclical activity in which: (1) memory in the form of schemata guides the exploration of the environment, (2) exploration yields samples of available information, and (3) data collected from the exploration process modifies the prevailing schema. By focusing on the interaction of perception and memory, Neisser's "perceptual cycle" model offers a particularly fertile context for studying the processsing of event information. However, because this is a processing model, rather than a model of knowledge representation, little emphasis is placed on the structure of schematized knowledge. Thus, it is not clear how *turning ignition* might be related to *driving home* or even what role the decomposition of events might play in generating the expectations purportedly used to guide sampling of available information. Germane to this issue is another that arises in relation to the proposed modification process. How does the prevailing schema change in response to the sampling process? In particular, what is the basis for the

† Also a member of the Cognitive Science Laboratory, Princeton University, 221 Nassau Street, Princeton, NJ 08540

similarity between the ongoing situation and the schemata that are subsequently activated?

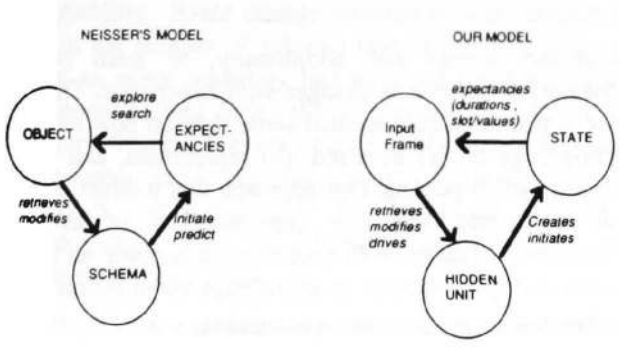

Figure 1: Neisser's Perceptual Cycle and Recurrent Net

The Problem with Scripts

Perhaps the best known attempt to address the kinds of questions raised here has been made by Schank (1982) within an artificial intelligence framework. Schank's approach to the parsing problem is essentially a taxonomic one in which relatively abstract knowledge structures (i.e., MOPs and TOPs) are posited to emerge from relatively specific action sequences (i.e., scripts). He suggests that comprehension emerges from a "reminding" process in which we "pass through old memories while processing a new input" (p.25).

"Reminding" is posited to occur when an online event activates an appropriate knowledge structure as a function of the similarity between the two. Thus, "reminding" is a process not unlike that posited in exemplar based categorization models (e.g., Medin & Schaffer, 1978) or the myriad of "nearest neighbor" algorithms posited to account for pattern recognition performance (Dasarathy, 1990). But, defining similarity remains as much a problem for Schank as for others wrestling with categorization issues.

Regrettably, similarity is invoked again when questions about structure development are raised. Structures at high levels in the hierarchy are posited to function as prototypes and to be abstracted from lower order structures. According to Schank (1982), these new high level structures develop "where the essential similarities between different experiences are recorded" (p. 81).

In addition to an inherent vagueness about the mechanism underlying the retrieval and development of knowledge structures, another problem with Schank's (1982) approach is its failure to deal with the temporal character of event knowledge in any straightforward way. Events persist for a given duration. Moreover, not only do different events persist for different durations, but the same event may last for different periods of time as a function of any number of factors such as the age of the actor, the experience of the actor, the time of day when the event takes place, the location of the event, and so on. An event not only persists for a given duration but derives its meaning from the context in which it occurs, that is, the events that precede and follow it. In itself, *ordering* means very little and *ordering* after *eating* makes little sense. It is only when *ordering* occurs in its rightful place among *sitting*, *eating*, and *leaving* that any real understanding can occur.

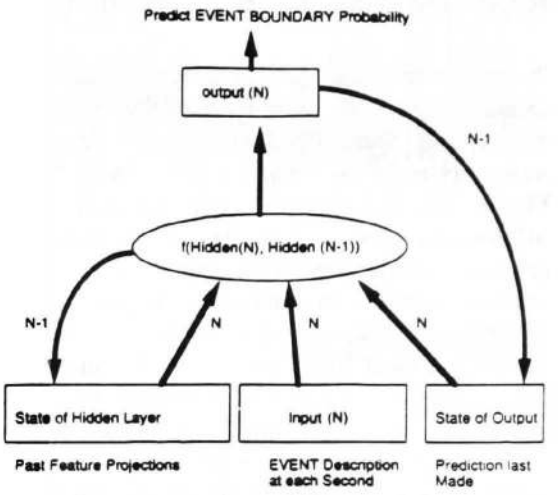

Figure 2: The Present Recurrent Net

Event Parsing

One way to avoid some of the difficulties that arise when a script structure is implemented is to model data derived from a task that creates context in terms of meaningful sequences of actions. That is, the nature of organizing schemata can be abstracted from human judgements concerning event boundaries for everyday situations. Data from a study by Hanson and Hirst (1989) provide such information. Briefly, subjects in this study were asked to watch videotapes of common event sequences. One videotape showed two people playing a game of Monopoly and the other showed a woman in a restaurant who drinks coffee and reads a newspaper.

TAPE	AGENT	ACTION	TYPEofVERB	OBJECT	MOVEMENT
game	mark	puts	transitive	money	yes
restaurant	pam	puts	transitive	money	yes

Figure 3: Input Encoding Example

Subjects watched the videotapes under various orientations and pressed a response button whenever they believed a new event was beginning. In the present study, we used reponses made when subjects had been oriented toward "small" events while viewing the tapes. This orientation produced the greatest number of perceived event boundaries and therefore a rich data set for use in training and transfer simulations.

Recurrent Nets and the Perceptual Cycle

A connectionist simulation provides an opportunity to examine how prior experience affects the parsing of actions into events. Recurrent networks, for example, inherently resemble Neisser's perceputal cycle [1] (See Figure 1). A recurrent net provides feedback information from hidden layers or from outputs creating information from either past actions at various moments in time or from past judgements about the presence or absence of an event change. For the net, an input frame consisting of a set of features and an arbtrary unit of time represents an object and moment of time in the world. The hidden layer of the network, which is driven by the input, also retrieves a learned category (Neisser's schema) which causes some moments in time to have a certain similarity to others (based on features). The feedback to the hidden layer creates a state (Neisser's expectancies) that influences in a top-down fashion judgements about the similarity of the present moment to an active schema retrieved via the hidden layer by the input frame.

A second reason for using a connectionist simulation is the opportunity it affords to examine the "black box" between input and output. By analyzing the hidden units of the network we hoped to gain some understanding about the kind of information needed to represent events and additionally, to learn how memory about events changes with experience. Thus, we hoped to be able to shed some light on how event knowledge is: (a) acquired, (b) represented, and (c) used to guide parsing. Our approach was a direct one; we examined how the representation of event knowledge changes with experience and observed the net's ability to transfer its knowledge about events to related and unrelated action sequences.

1. Rumelhart first suggested this connection between the perceptual cycle and reccurrent nets.

Network Structure and Training

A simple recurrent network used sources of information including features of events from the present moment in time, past event-moment features and past predictions of an event change (see Figure 2). It is known that simple recurrent networks (Elman, 1988; Rumelhart, Hinton & Williams, 1986) can represent *at least* a finite state machine (Servan-Schreiber, Cleremans & McClelland, 1988; Watrous & Kuhn, 1991; Giles et al., 1991) and thus are good candidates for encoding temporal event sequences. The present recurrent network received feedback from hidden layers and outputs delayed by one time step. Inasmuch as these activation values were combined over time they potentially can represent a complete sequence from the start of the event parsing.

Input Encoding. As stated before, two kinds of videotaped action sequences were used as data, one involving two people playing a Monopoly game and another involving two people in a restaurant sequence. Each tape was transcribed to the resolution of one *second*. Five variables were chosen to represent each second of the event sequences. These variables included AGENT, ACTION, OBJECT, TYPE of VERB (transitive or intransitive) and MOVEMENT (whether any movement occurred in that second). In Figure 3 are examples of a single second transcribed for each kind of videotape: Hereafter, this attribute-value structure will be referred to as the frame-second. The combined information from both tapes included a total of 4 AGENTS, 33 ACTIONS and 43 OBJECTS. Sixty percent of ACTIONS and 9% of OBJECTS overlapped between the two event sequences. The

network was provided a binary representation (17 bits) of this input frame.

Training. The network's task was to learn to map the current frame-second, any past frame-second, and the past event change probability to the next event change probability. Event change probability was computed from the number of subjects (out of 20) who judged that an event boundary had occurred (by pressing a response button) during that frame-second. Shown in Figure 4 are the event change probablities[2] for the Monopoly game. On the x-axis are the 420 frame seconds corresponding to the transcribed features. The y-axis shows the relative frequency of button presses at that second.

The network was trained by 1st-order gradient descent ("back-prop in time") to produce the event change time series. Due to the noise present in the time series other methods such as line-minimization or conjugate gradient methods (e.g. BFGS optimization) fared poorly in terms of speed of convergence and reliability to the same solution as a function of starting point. Simple 1st-order back-prop, converged quickly and reliably to the same solution in spite of the target noise.

Standard Models. The event change probabilities were not modelled well as an ARIMA (Box-Jenkins) time-series suggesting few periodicities were present in the time-series independent of the frame-seconds. A standard multiple regression accounted for less than 5% (Pearson r correlation of .07) of the data variance suggesting that the mapping was significantly nonlinear.

Learning of Game and Restaurant Tapes

Using split halfs of the Game and Restaurant tapes the recurrent network was able to acount for over 45%-50% of the data variance with Pearson r correlations of .75 for the Game and .68 for the Restaurant tape. The difference appears to be related to the differences in length of the tapes (Game-420 seconds, Restaurant-287 seconds) and the higher diversity of actions in the Game event sequence.

2. Subsequent figures of event change probabilities show a smoothed (4-second window) version of this data in order to make visual comparisons only. All prediction values from network training are for the raw data shown in Figure 4.

Transfer. The Game tape learning was then transferred to the Restaurant tape[3]. Shown in Figure 5 in the left panel is the result of training on the Game tape. The dashed line represents the event-change data and the solid line is the second-by-second prediction of the reccurrent network. Using all the data for the Game tape boosted the variance accounted for to 80% (r=.9). Transfer to the Restaurant tape, was significant (40%, r=.65) in spite of large attribute-value differences at each second of each tape. Hidden unit sensitivity was explored for 5 to 30 hidden units. Variance accounted for on transfer went up slowly, reaching asymptote near 15-20 hidden units.

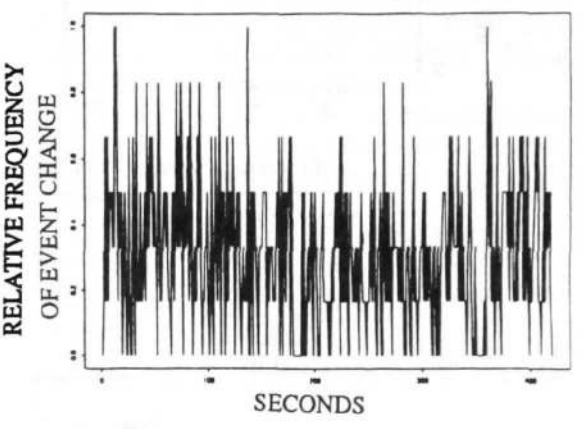

Figure 4: Event Change Judgements over Time (seconds)

Internal Representation

Hidden unit patterns were analyzed (Hanson & Burr, 1990) over each second in order to determine the similarity of frame-seconds that the reccurrent net discovered to make the event change predictions. A hierarchical cluster analysis (Centroid, and Farthest Neighbor agreement) was performed on the hidden unit activations over the 420 seconds and over the 287 seconds. Very regular dendrograms were produced and an examination of successive differences over the merge history indicated 10-15 clusters to be present.

Insofar as clusters represent groups of frame-seconds that are similar from the reccurrent net's point of

3. The Restaurant tape was also tranferred to the Game tape, but with less success, probably because the sample size of the restaurant tape was about 75% that of the Game tape.

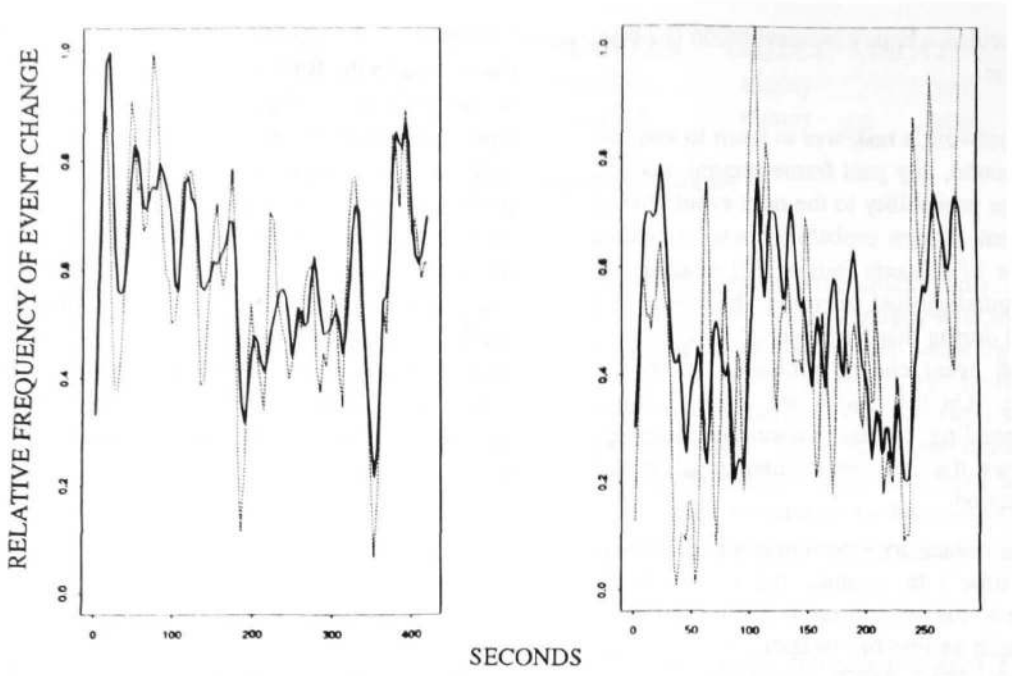

Figure 5: Transfer from the Game to the Restaurant Event Sequence

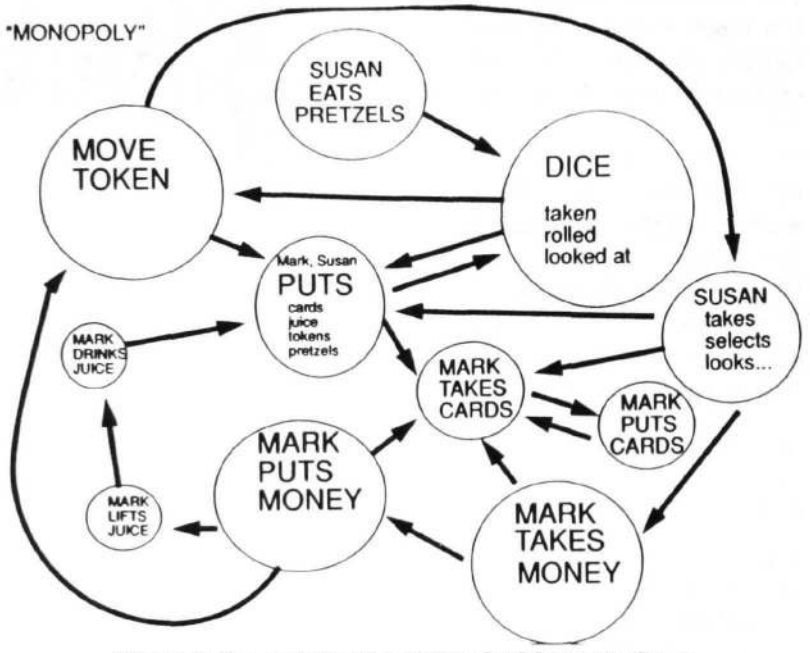

Figure 6: Internal Representation for Monopoly Game

view, each cluster was identified as a schema, and used to relabel the sequence of frame-seconds in each tape. For example, the 420 seconds of the Game tape was relabelled with the 11 identified clusters or schemata. A graph of the new sequence, with the schemata labelled using the common features of each frame second (e.g., if PUTS was the only common feature in the cluster the schema was labelled "PUTS", if MARK TAKES MONEY was common to every frame-second then the schema was labelled "MARK TAKES MONEY"), is plotted in Figure 6.

The size of each ball is based on the relative frequency of the schema in the tape, and the arrows represent the state transition (a state transition would predict an event change for the recurrent net). Note that the sequence of Monopoly events is representated in this graph, and that different schema level abstractions have resulted as a function of learning the event change probabilities. Some schemata represent information at the exemplar level whereas others have generalized by dropping AGENT or ACTION or OBJECT or subsets of these variables. Finally, notwithstanding the differences in frame-second content (especially in terms of objects) and poorer prediction performance, a representation was extracted that did correspond to action sequences consistent with the actual events in the Restaurant sequence.

The Nature of Event Perception

We conclude by providing answers to the questions about schemata posed earlier in this paper. Based on the computation of the recurrent net and the internal representations of schemata extracted, several aspects of event perception might be clarified.

On what basis are schemata activated? Several factors determine whether a schema is activated or not. One, similarity (in this case dot-product to the hidden layer) of a schema to a present frame-second (in terms of attribute and value presence) can activate and retrieve new schemata in a bottom-up fashion. Two, past schemata will resist bottom-up input at a given frame-second and will tend to block the activation of new schemata. The more specific a schema is (in terms of attributes and values) the less likely transitions to a new schema will occur. Three, each schema has been associated with an expected duration. The duration of a schema can be determined by clamping the plan vector with a given attribute value and starting input values at ambiguous values (.5) and counting the number of seconds passed before the output approaches a value between .75 and 1.0. All 11 schemata for the Game tape were clocked in this way. If a schema is expected to continue for a long time, inconsistent input data will be ignored until a sufficient number of instances appear. Some schemata occur frequently having brief durations while others occur rarely but at longer durations. In fact, there was a significant negative correlation (-.52) between schema duration and frequency.

What role do schemata play during encoding? The active participation of schemata help to select input and maintain resistance to change. Within the context of the recurrent nets, schemata create expectations about the level of abstraction that will appear in the input frame and the specific content that should be found. Finally, once a schema is activated there is an expectation about its duration (due to the feedback) and a search for confirmation continues until the schema terminates.

How do schemata develop? The frequency of events and their duration within the frame-seconds determine how schemata develop and what properties they will possess. As stated above, there is an inverse relation between duration of schemata and their frequency in the Game tape. High-frequency, short-duration schemata tend to be more asbtract or general, whereas low-frequency, long-duration schema tend to be more specific or exemplar based. Examination of the schemata as they develop during learning indicate that they tend to evolve from specific exemplar based clusters into more abstract based clusters by accepting increasingly more diverse input over time.

References

Bower, G.H., Black, J.B., & Turner, T.J. (1979). Scripts in memory for text. *Cognitive Psychology*, 11, 177-220.

Dasarthy, B. (1990). NN Pattern Classification Techniques, Los Alamitos, CA: IEEE Computer Society Press.

Elman, J.L. (1988). *Finding structure in time*. CRL Technical Report 8801. Center for Research in Language, UCSD.

Giles, L., Miller, C. B., Chen D. Chen, H. H. Sun G. Z., Lee, Y.C. (1992). Learning and Extracting Finite State Automata with Second-Order Recurrent Neural Networks, *Neural Computation*, (in press).

Hanson, C. & Hirst, W. (1989). On the representation of events: A study of orientation, recall, and recognition. *Journal of Experimental Psychology: General*, 118, pp. 124-150.

Hanson, S.J. & Burr, D. J. (1990). What Connectionist Models Learn: Learning and Representation in Connectionist Networks. *Behavioral and Brain Sciences*, 13, 3 pp. 477-518.

Medin, D.L., & Schaffer, M.M. (1978). Context theory of classification learning. *Psychological Review*, 85, 207-238.

Neisser, U. (1976). *Cognition and reality: Principles and implications of cognitive psychology*. San Francisco: W.H. Freeman.

Newtson, D. (1973). Attribution and the unit of perception of ongoing behavior. *Journal of Personality and Social Psychology*, 28, 28-38.

Rumelhart, D., Hinton, G. & Williams, R. (1986). Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland (Eds.), *Parallel Distributed Processing I: Foundations*. Cambridge, Mass: MIT Press.

Schank, R.C. (1982). *Dynamic Memory: A theory of reminding and learning in computers and people*. Cambridge: Cambridge University Press

Servan-Schreiber D., Cleeremans, A. & McClelland, J. (1988). *Encoding sequential structure in simple recurrent networks*. CMU Technical Report CS-88-183.

Watrous, R. & Kuhn G. (1992). Induction of Finite -State Languages Using Second -Order Recurrent Networks, *Neural Computation*, (in press).