The Story Gestalt

Text Comprehension by Cue-based Constraint Satisfaction

Mark F. St. John

Department of Cognitive Science
University of California at San Diego
La Jolla, CA 92093[1]

How do readers comprehend text? How do they build a representation of the event to which the text refers? How do they infer important causal information that the text does not explicitly mention, and how is information from different past experiences combined and modified to fit the current context? For example, how do readers comprehend, "Jolene raced down the track ahead of her rivals. At the finish line, the judge handed her the trophy," and infer that Jolene won the race?

A new algorithm, called cue-based constraint satisfaction or CCS, provides a different way of understanding a number of important processes in text comprehension. In CCS, the text is viewed as consisting of a set of cues that provide evidence for an interpretation. This evidence is represented as real-valued constraints on a successful interpretation of the text. Constraints are easily combined, and are used to compute an interpretation of the text.

The goal of this paper is to present this new algorithm and demonstrate its approach to performing a set of comprehension processes. The processes addressed are representing the explicit text, resolving pronouns, drawing coherence and prediction inferences, revising the interpretation of the text as new information is processed, and sharing knowledge learned in different contexts.

In more traditional models, such as Charniak (1983), a sentence is processed into a proposition, then a semantic network is traversed to find a path connecting the proposition to

[1]e-mail: stjohn@cogsci.ucsd.edu

previous text. Concepts along the connecting path constitute inferences drawn to make a coherent interpretation. Alternately, as in Schank's (1981) model, a detailed script is retrieved from memory and the text is mapped onto it. In both of these approaches, processing the explicit text, resolving pronouns, and drawing inferences work separately. Information is represented in an all-or-none fashion that does not reflect the gradedness of information in the real world. For example, the likelihood of a causal relationship is not represented. Third, revision is difficult because changes to an all-or-none representation are necessarily drastic. Consequently, these systems are conservative and rarely make predictions or other risky inferences that may require revision.

Recently, a number of models closer to the idea of constraint satisfaction have been developed (Kintsch, 1988; Miikkulainen & Dyer, 1989). The algorithm and model presented here is a further step toward articulating a theory of comprehension by constraint satisfaction and providing a more functional model of the algorithm. Plus, the model presented here learns its representation of stories from experience. By having the model learn for itself, the modeler is freed from the difficult, arbitrary task of programming the representations and knowledge. Further, our understanding of what is required from a corpus of experiences and the training task posed for model to achieve good performance may be enhanced.

To begin, the processing tasks are briefly introduced. Then the model is described. The CCS algorithm can easily be implemented in a parallel distributed processing network that sequentially processes the propositions explicit in the text into a representation of the meaning of the text that includes important inferences. Finally, simulation results are reported that demonstrate the algorithm's competence.

Comprehension Tasks

Representing multiple propositions. A basic requirement is the ability to represent multiple propositions in such a way that their constraints on the story interpretation can be combined and used effectively. Each proposition built from the text provides constraints on the characters and actions in the story. The combined constraints from all of the propositions provide the evidence to support interpretations.

When the constraints are combined, though, it is important that they not become confused. In "Fred gave the book to Barney," the fact that it was Fred who gave the book, and not Barney, must be retained. In some models, however, these role assignments are lost during the vary act of comprehending, and expensive additional processing is required to recover them (Charniak, 1983).

Resolving pronouns. Pronouns create substantial ambiguity in text. When referring to people, they constrain only the gender and number of their referents, which may or may not be helpful. Additional constraints come from the referential situation. For example in, "Fred picked up the frisbee. He threw it to Barney," The text supports an interpretation in which Barney is the recipient of the throw. This interpretation also represents Fred as the agent; so the pronoun "he" must correspond to Fred. Similarly, series of actions involving one character can provide constraint. If the identity of the character is known from one action, pronouns found in the remaining actions can be resolved.

Finally, the referent for a pronoun may remain ambiguous for some time. The model should be able to tolerate this ambiguity and then combine whatever information later becomes available to resolve the pronoun.

Inferring propositions. Inferred propositions can be divided into two categories. Coherence inferences are drawn to explain or justify the text, and prediction inferences are drawn to predict additional information, such as future actions, that fit the context.

Empirical evidence suggests that prediction inferences are drawn, but only activated weakly according to the amount of support provided by the text (Graesser, 1981; McKoon & Ratcliff, 1986; but see Potts, Keenan, & Golding, 1988.). Coherence inferences, on the other hand, are fully activated because of their stronger support from the text. Models of text comprehension should consequently incorporate a mechanism that uses the degree of support provided by the text to determine the activation level of inferences.

Revising an on-going interpretation. Readers revise their interpretations when new information makes their old interpretations unlikely. Rumelhart (1981) asked subjects questions about their interpretation as they read the following story.

> *Business had been slow since the oil crisis. Nobody seemed to want anything really elegant anymore. Suddenly the door opened and a well dressed man entered the showroom floor. John put on his friendliest and most sincere expression and walked toward the man.*

Rumelhart found that after reading the first sentence, his subjects held a variety of interpretations. As additional sentences were read, his subjects changed their interpretations. This process of evaluation and revision is a ubiquitous aspect of comprehension.

Learning. Finally, the model's knowledge should be learned from examples. Learning is difficult in may conventional symbolic systems because arbitrary decisions must be made, for example, about when to encode an exceptional story as an exception within a script or as a distinct script. Under the CCS view, learning is simpler. Since constraints are graded, these decisions do not have to be made: the exceptional story is encoded by strengthening or weakening a large number of constraints.

It can be argued that scripts and frames are a rough approximation to the nature of the events they represent. The all-or-none boundaries they impose on the representation of events are artificial (Rumelhart, Smolensky, McClelland, & Hinton, 1986). The true nature of common events is better described as a set of prototypes that develop from exposure to numbers of instances (McClelland and Rumelhart, 1985; and Posner, and Keele, 1968). The instances encode all of the salient details. Prototypes emerge from the correlations between details. The prototypes most similar to a new instance will make the strongest predictions, and the strength of the correlations between details will determine the strength of the predictions: there is no need to make binary discriminations between script-relevant and script-irrelevant details.

A successful paragraph comprehension mechanism, therefore, should be able to encode instances and extract the correlations among their features. The constraints that encode the correlations can then be applied to the comprehension task.

Simulation Method

The model's task is to take a story as input and understand the story so that it can answer questions. The model learns to comprehend stories through experience comprehending example stories. Once trained, the model's performance on each task can be evaluated by using example stories from the corpus or by using novel stories.

Corpus. The corpus consists of a large number of stories that take place in one of six different contexts: for example, going to the beach or going to a restaurant. Each story consists of a series of propositions that describe actions or attributes of a character or of the situation.

A proposition is represented as a set of thematic roles: agent, predicate, patient or theme, recipient or destination, location, manner, and attribute. For example, the proposition that conveys that *the judge gave the trophy to Jolene*, would be (agent=judge, predicate=gave, patient=trophy, recipient=Jolene, location=nil, manner=nil, attribute=nil). The "predicate" role is used both for actions and for the predicate of descriptions. For example, the proposition for *it was sunny at the beach*, would be (agent=nil, predicate=weather, patient=nil, recipient=nil, location=beach, manner=nil, attribute=sunny).

A proposition is represented in the network as a vector of activation values. The concepts are represented locally, so that each possible concept for each thematic role is represented by a separate, single unit. Consequently, there are 19 units to represent each of the 19 agents. There are 34 different units to represent each of the 34 patients, and so on. A *nil* for a role is represented by turning off every unit in that role. All together, 136 units are required to represent all of the propositions.

The corpus of stories is generated from a set of six story-frames. Each story-frame contains information about the events that can occur in one of the six contexts. This information is chosen probabilistically to be included in a particular story. The model is trained on the corpus by first generating a story from one of the story-frames and then training the model on that story. Both a story and the event it refers to are generated at the same time. The story may leave out information and contain pronouns, but the event is always complete and specific.

Across the six story-frames, the total number of different events is 28,480. For example, in the restaurant story-frame, there are 20 different sequences of events involving two of ten possible characters and one of four possible vehicles for a total of 7200 restaurant events (20 * 10 * 9 * 4). The number of possible input stories is much higher because of pronouns and missing propositions.

Architecture and Training Regime. To comprehend one of these stories, the model processes the propositions one at a time, and evaluates the information in each proposition to iteratively build and refine a representation of the whole story. The network is trained to perform this task by receiving feedback on its comprehension performance. In essence, the model is required to process the propositions so that it can answer questions about them. The questions provide a predicate from a proposition in the event and the network is required to complete the whole proposition from the predicate alone.

The network can be divided into two parts according to their function. Part A of the network (see Figure 1) receives each proposition in turn and processes it to update its representation of the story. Specifically, the proposition is used to activate the units in the *current proposition* layer. The *current proposition* layer sends activation to the *intermediate-combination* layer where it is combined with activation from the *previous story gestalt*. The story gestalt represents what is known about the story at each point during processing. At the beginning of the story, nothing is known, so the units all have 0.0 activation. Activation from the *intermediate-combination* layer feeds forward to produce a pattern of activation in

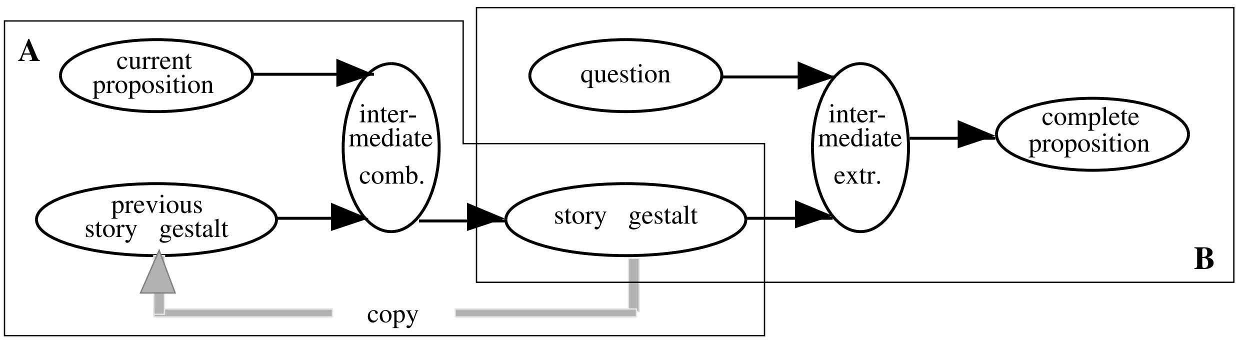

Figure 1. The architecture of the Story Gestalt network. The boxes highlight the functional parts: Area A processes the propositions into the story gestalt, and Area B processes the story gestalt into the output propositions. See the text for details.

the *story gestalt* layer. The new *story gestalt* then represents the whole story as well as it can based on the information provided from the input at that point in the story.

As each new proposition is processed, the new proposition replaces the last one in the *current proposition* layer, and the pattern of activation over the *story gestalt* is copied to the *previous story gestalt* layer. Activation feeds forward again to create a new story gestalt. In this way, information from previous propositions is both maintained and is used to compute an updated interpretation of the whole story. It is important to understand that the story gestalt representation need not be simply a weighted sum of the propositions. The two layers of weights between the *previous story gestalt* and the *story gestalt* allow the network to compute more complex and useful representations of the information contained in the propositions.

Part B of the network performs the question answering computation. A question is created by removing everything except the predicate from a proposition. The question is then used to activate a pattern of activation in the *question* layer. Activation from this layer combines in the *intermediate-extraction* layer with activation from the *story gestalt*. Activation from the *intermediate-extraction* layer then activates the units in the *complete proposition* layer.

There are 136 units in the *current proposition* layer, 100 units in the *story gestalt* and *previous story gestalt* layers, 100 units in the *intermediate-combination* and *intermediate-extraction* layers, 34 units in the *question* layer, and 136 units in the *complete proposition* layer.

The propositional representation used in the *complete proposition* layer is identical to the representation used in the *current proposition* layer. The activation pattern over the units in the *complete proposition* layer can be compared to the correct pattern for that question. The error between the two is back propagated (Rumelhart, Hinton, and Williams,

1986) to the *current proposition* and *previous story gestalt* layers. The error is calculated using the cross-entropy function (see Hinton, 1989), where T_j is the target activation and A_j is the output activation of unit j.

$$C = -\Sigma_j [T_j * \log_2(A_j) + (1-T_j) * \log_2(1-A_j)]$$

The model is trained after each proposition on the part of a story that has been presented in the input. After the first proposition, the model is asked only about the first proposition. After the second proposition is presented, the model is asked about both the first and second propositions. Propositions that are missing from the input are only asked about after they have been skipped in the input. For example, if the second proposition were missing, the model would only be asked about it after the third proposition were presented.

Before training on the corpus began, the model was trained on single, random propositions. The propositions are generated by randomly activating one unit in each thematic role, and the model is required to reproduce that proposition in the output. The purpose of this training is to help the model learn to remember the propositions actually presented by learning to map single propositions from the input to the output. Once this mapping has been learned, the model can then be trained on whole stories. Training, then, is composed of two phases: a pretraining phase on single propositions and the normal training phase on whole stories.

Once the model has mastered the pretraining task, training on the full corpus of stories begins. So that the model will not immediately lose the memory mapping it learned in the pretraining, the random, single proposition training continues on 20% of the training trials.

Simulation Results

The model was pretrained on the single propositions until the total sum squared error across the output fell below .05. This training itself required 145,000 training trials with a learning rate of .001. The model was then trained on stories for 160,000 trials (Numbers of trials here do not include trials on the random, single propositions.) with a learning rate of .0001 and a momentum of .9, then for 20,000 more stories with a momentum of 0. The model's performance was then evaluated.

Figure 2 demonstrates several phenomena. A story is presented to the model that contains pronouns and that skips a number of propositions. The model must resolve the pronouns correctly and infer the missing information.

The corpus of restaurant stories contains a number of regularities, that if learned, can be used to help perform these tasks. One such regularity is that the character who orders also pays the bill and receives back her credit-card. Given which character performed any of these tasks, the model can use the regularity to infer the identity of the character in each of the other tasks. As the example shows, given that Sally paid the bill, the model is able to infer that she also ordered and received back her credit-card. In effect, the model has assigned Sally to the "payer" script role and has instantiated the character in each task to be Sally.

Another regularity that can aid inference is that, in the corpus, expensive restaurants are always far away and cheap restaurants are always nearby. If the model has learned this regularity, it can use the story information that the restaurant is expensive to infer that it is far away. As the example shows, the model can draw this conditional inference.

Finally, the model can draw inferences based on a previous experience with particular characters. The model uses previous experience with Sally to infer that she will order cheap wine.

input

(agent=Sally and Clement, predicate=decided to go, destination=restaurant)
(predicate=quality, patient=restaurant, attribute=expensive)
(agent=Sally, predicate=paid, patient=bill, location=restaurant)
(agent=she, predicate=tipped, recipient=waiter, location=restaurant, manner=small)

question	**output**				
	agent	**predicate**	**destination**		
decided to go	S and C	decided to go	restaurant		
	9 9	9	9		
	predicate	**patient**	**attribute**		
quality	quality	restaurant	expensive		
	9	9	9		
	predicate	**patient**	**attribute**		
distance	distance	restaurant	far		
	9	9	9		
	agent	**predicate**	**patient**	**recipient**	**location**
ordered	Sally (C)	ordered	cheap wine	waiter	rest.
	8 (5)	9	9	9	9
	agent	**predicate**	**patient**	**location**	**manner**
enjoyed	S and C	enjoyed	cheap wine	restaurant	not
	9 9	9	9	9	9
	agent	**predicate**	**patient**	**location**	
paid	S (C)	paid	bill	restaurant	
	9 (1)	9	9	9	
	agent	**predicate**	**patient**	**recipient**	
brought	waiter	brought	credit-card	Sally (C)	
	9	9	9	7 (4)	
	agent	**predicate**	**patient**	**location**	**manner**
tipped	Sally (C)	tipped	waiter	restaurant	small
	9	9	9	9	9

Figure 2. Conditional inferences. The model is shown four propositions from a restaurant story, then questioned about a number of propositions from the whole event. The numbers indicate the activation, from 0 to 1.0, of active concepts. The parentheses denote alternative concepts and their activations. When there are two agents, the activation of each is shown.

Next, the model's ability to revise it's on-going interpretation of a story was tested. In the bar story, there are two pairs of propositions that predict the final proposition of the story. Before either proposition is presented, rubbed lipstick and rubbed cheek are equally likely. The first pair predicts the final proposition probabilistically. A polite pass predicts rubbed lipstick 70% and rubbed cheek 30%, while an obnoxious pass predicts rubbed lipstick 30% and rubbed cheek 70%. The second proposition, or cue, predicts the final proposition absolutely: gave kiss predicts rubbed lipstick 100% and gave slap predicts rubbed cheek 100%.

With no information from either cue, the model should activate both possibilities partially. After the pass proposition, the activation of lipstick and cheek should reflect the probabilistic evidence from the first cue. Since the second cue is completely predictive, the model should fully activate the outcome that fits with that cue: lipstick for kissed, and cheek for slapped. Table 1 first shows the *ideal* activations of lipstick and cheek given the above analysis. The *nils* in the table indicate conditions where one or the other of the cues is unknown.

The *observed* activations in Table 1 are averages from three test cases per cell. As table 1 shows, before the first cue is presented, the predictions for the final proposition should be equal. The model, however, shows a bias for one alternative. This bias is presumably due to noise in training. After the probabilistic first cue is presented, the model should activate each alternative according to the probabilities each alternative will come true. The model's activations correspond roughly with the ideal activations.

Organization of Bar stories

initial prop.:	Person_1 decided to go to the bar.
first cue:	Person_1 made a polite/obnoxious pass at person_2.
second cue:	Person_2 gave a kiss/slap to person_1.
final prop.:	Person1 rubbed lipstick/cheek.

Table 1
Successive Revision in Bar Stories

			ideal				
first cue:	nil		polite			obnoxious	
second cue:	nil	nil	kiss	slap	nil	kiss	slap
object							
lipstick	5	7	9	0	3	9	0
cheek	5	3	0	9	7	0	9

			observed				
first cue:	nil		polite			obnoxious	
second cue:	nil	nil	kiss	slap	nil	kiss	slap
object							
lipstick	7	6	7	2	3	6	1
cheek	2	2	1	5	5	2	7

After the second cue is presented, the model revises its interpretation to reflect the new information. The model moves the activations in the appropriate direction, and the correct response is more active than the incorrect response in each case. The model, therefore, represents and revises its interpretation based on the reliability of evidence gleaned from the story.

Finally, how might novel stories be understood? Under the constraint satisfaction view, the comprehension of novel stories simply consists of combining novel sets of constraints to produce an interpretation of a novel story. The constraints on a story interpretation can be divided into two categories: memory constraints and associative constraints. Memory constraints represent the correlations of information with itself, and they are responsible for the recall of the information explicitly mentioned in the text. Associative constraints represent the correlations between pieces of information, and they are responsible for inferences and pronoun resolution.

For novel stories, memory constraints cued from each part of the story are used to activate parts of the story gestalt. The difficulty for novel stories is that each cue, through its associative constraints, actually constraints many parts of the story in context specific ways. When the cues are placed in new combinations, many of these correlated parts of the story interpretation will contradict each other. The difficulty, then, is keeping the associative constraints, that activate the correlated parts, from overriding the memory constraints.

These ideas were tested in a new corpus pared down to the essential details. First, a corpus was designed to demonstrate the ability of strong associative constraints to override information in novel stories. This corpus contained only two stories: one about an expensive restaurant and one about a cheap restaurant. The stories were four propositions long. Each proposition contained a role that could take on one of two values. The cheap restaurant had the values cheap, near, change, and small. The expensive restaurant had the values, expensive, far, credit-card, and large.

Organization of New Restaurant Stories

predicate=quality, patient=restaurant, attribute={cheap, expensive}
predicate=distance, location=restaurant, attribute={near, far}
agent=Andrew, predicate=paid, manner={change, credit-card}
agent=Andrew, predicate=tipped, manner={small, large}

Once trained, the model was tested with stories containing combinations of role values the model had not seen: e.g. an expensive restaurant where Andrew pays with change. The model strongly overrides change and activates credit-card. The associative constraints do not allow the irregular role value to be activated by the much weaker memory constraint from change.

A fresh model was then trained on an expanded corpus. The expanded corpus again contained the two restaurant stories. In addition, it contained stories in four other locations. Each of these stories again consisted of the same four propositions, except that the context was changed to the new location. In each of these four new locations, all 16 of the possible combinations of role values (2^4) were trained. With this new corpus, the model cannot rely on associative constraints to predict one part of a story from the others, because there are no correlations between role values. Instead, the model must learn strong memory constraints to be able to comprehend these stories. The hope is that these stronger memory constraints will be able to override the associative constraints in the restaurant context, and allow the novel stories to be recalled correctly.

Once trained, the model was again tested with the expensive-restaurant-change story. The model now can represent the story despite never having seen that combination of role values before, and despite the strong regularities in the restaurant context fighting against that combination. The memory constraints win, but the activation of change is reduced due to the competition from the associative constraints. Interestingly, if the proposition about paying is missing, the associative constraints are applied, and the model infers credit-card, the regular value.

What the model has learned from this corpus, then, is something like a variable slot that has a default value. The model will recall the actual input, but if the input is missing, it will infer the default. However, the default is felt even when a value is provided in the input. What the model has developed, then, are "graded slots" where constraints compete according to their strengths.

Of course sharing knowledge runs in both directions. While the restaurant context gains the use of the memory constraints, the other contexts gain the constraints among role values. When a value is missing in one of these context, the script-appropriate default value from the restaurant context is inferred. So, in an expensive beauty salon where the tip was large, the model will infer that a credit-card was used.

Conclusions

Several of the model's specific features are problematic. One problem is that the proposition representation is rather clumsy. First, each story can use a particular predicate only once because the predicates must be used as unambiguous questions for training. Second, the seven thematic roles are both arbitrary and confining. Ideally, the model would learn its own representation for propositions just as it learns its own representation for story gestalts.

Another problem is that learning is very slow due, in part, to the depth and size of the network. The error must be passed back through several layers of weights, and each layer contains a large number of weights, so credit assignment is difficult.

A larger issue here is that the model has no intrinsic understanding about what can and cannot occur together in a story. It does not intrinsically know that you cannot be at a beach and an airport at the same time. The model can learn this knowledge only from the fact that it never occurs in its training corpus.

In spite of these difficulties, the model performs well on the comprehension tasks and demonstrates the cue-based constraint satisfaction approach. CCS brings several new ideas to our understanding of text comprehension. The most basic idea is that the interpretation of the text is supported by constraints from the text rather than build from it by appending successive propositions to some structure. The constraint view carries with it the idea that resolving pronouns, drawing inferences, and representing the explicit text result from the same process: the constraining of an interpretation. Each constraint supports or inhibits features of the story gestalt: it constrains what the gestalt will represent. The gestalt that results from applying all of the constraints is the one that satisfies these constraints the best. Inferences and pronoun resolution occur when one part of the story provides constraints for other parts that were underconstrained in the story. This process is more sophisticated than simple pattern completion since it is capable of binding characters to their roles in a script.

A second idea is that these constraints can be graded, and their support of an interpretation can be graded. Pieces of text often may be only partially reliable as cues for an interpretation. A polite or obnoxious pass, for instance, is only partially reliable at predicting the outcome in the bar stories. Despite its inconsistency, it is valuable information. It has some predictive value and could potentially be combined with other information to produce a good interpretation. Allowing constraints to take on strength values according to their reliability, therefore, allows the model to use less than completely reliable information to its advantage.

The constraint view also carries with it the idea that each constraint makes its own contribution to the interpretation. Each new constraint adds its own support to the evolving interpretation. This independence of constraints allows the model to update its interpretation as each new constraint is processed.

Additionally, the idea of independent constraints shows how novel stories can be understood. Each constraint adds its own support and influence to the whole interpretation. Understanding a novel story involves combining the memory constraints from each part of the input. The associative constraints from each part add extra information, but they may support conflicting information. Misremembering a story to better fit prior experience can be one result of conflicting associative constraints (Bartlett, 1932).

Finally, the constraints imposed by cues can be learned. The graded strengths of constraints allows them to be learned incrementally with experience. The model does not have to decide all-or-none whether a constraint is valuable. It simply assigns a strength to that constraint commensurate with its reliability.

To sum up, cue-based constraint satisfaction provides a new approach to story comprehension. Some of the difficulties found in other models of story comprehension are ameliorated by the qualities of this approach. There is much to story comprehension that the model does not yet do, and the model faces some difficulties in learning speed and architecture, but the fundamental characteristics of the approach are successful and promising.

References

Bartlett, F. C. (1932). *Remembering: An experimental and social study.* Cambridge: Cambridge University Press.

Charniak, E. (1983). Passing markers: A theory of contextual influence in language comprehension. *Cognitive Science*, 7, 171-190.

Graesser, A. C. (1981). *Prose comprehension beyond the word.* New York, NY: Springer-Verlag.

Hinton, G. E., (1987). Connectionist learning procedures, technical report #CMU-CS-87-115, Computer Science Department, Carnegie-Mellon University.

Kintsch, W. (1988). The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review,* 95, 163-182.

McClelland, J. L. & Rumelhart, D. E. (1985). Distributed memory and the representation of general and specific information. *Journal of Experimental Psychology: General,* 114, 159-188.

McKoon, G. & Ratcliff, R. (1986). Inferences about predictable events. *Journal of Experimental Psychology: Learning, Memory, and Cognition,* 12, 82-91.

Miikkulainen, R. & Dyer, M. G. (1989). A modular neural network architecture for sequential paraphrasing of script-based stories. Technical report UCLA-AI-89-02, Artificial Intelligence Laboratory, Computer Science Department, University of California, Los Angeles, CA.

Posner, M. I., Keele, S. W. (1968). On the genesis of abstract ideas. *Journal of experimental psychology,* 77, 353-363.

Potts, G. R., Keenan, J. M., Golding, J. M. (1988). Assessing the occurrence of elaborative inferences: Lexical decision versus naming. *Journal of Memory and Language,* 27, 399-415.

Rumelhart, D. E. (1981). Understanding understanding. In H. W. Dechert and M. Raupach (Eds.), *Psycholinguistic models of production.* Norwood, NJ: Ablex Publishing.

Rumelhart, D. E., Smolensky, P., McClelland, J. L., & Hinton, G. E. (1986). Schemata and sequential thought processes in PDP models. In D. E. Rumelhart, J. L. McClelland, and the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition, Volume 2.* Cambridge, MA.: MIT Press.

Schank, R. C. (1981). Language and memory. In D. A. Norman (Ed.) *Perspectives on cognitive science.* Norwood, NJ: Ablex Publishing.