The Story Gestalt: A Model of Knowledge-Intensive Processes in Text Comprehension

MARK F. ST. JOHN

University of California, San Diego

How are knowledge-intensive, text-comprehension processes computed? Specifically, how are (1) explicit propositions remembered correctly, (2) pronouns resolved, (3) coherence and prediction inferences drawn, (4) on-going interpretations revised as more information becomes available, and (5) information learned in specific contexts generalized to novel texts? A constraint satisfaction model is presented that offers a number of advantages over previous models: Each of the previous processes can be seen as examples of the same process of constraint satisfaction, constraints can have strengths to represent the degrees of correlation among information, and the independence of constraints provides insight into generalization. In the model, propositions describing a simple event, such as going to the beach or a restaurant, are sequentially presented to a recurrent PDP network. The model is trained through practice processing a large number of example texts and answering questions. Questions are predicates from propositions explicit or inferable from the text, and the model has to answer with the proposition that fits that predicate. The model learns to perform well, though some processes require substantial training. A second simulation shows how the combinatorics in the training corpus can increase generalization. This effect is explained by introducing the concept of identity and associative constraints that are learned from a corpus. Overall, the model provides a number of insights into how a graded constraint-satisfaction model can compute knowledge-intensive processes in text comprehension.

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? In general, how are knowledge-intensive text processes computed?

A variety of models have been proposed to answer different sets of these issues (Charniak, 1983; Kintsch, 1988; Miikkulainen & Dyer, 1991; Schank & Abelson, 1977; Shastri, 1988, to name a representative five). These models span a large space of possible mechanisms. An analysis of their successes and problems suggests that graded constraint satisfaction (Rumelhart, Smolensky, McClelland, & Hinton, 1986; see also Bates & MacWhinney, 1987) offers good answers to how knowledge-intensive text processes are computed: The explicit text places constraints on an interpretation, and a constraint-satisfaction process computes the interpretation that best satisfies these constraints. According to this approach, representing the explicit text, resolving pronouns, and drawing inferences are part and parcel of the same process of constraint satisfaction. Second, because constraints can be graded, degrees of likelihood and strengths of relationships can be represented and used. Third, the gradedness of the representation makes the revision and refinement of interpretations easier.

The most comprehensive of previous graded constraint-satisfaction models, both in terms of processing and learning, is the Miikkulainen and Dyer (1991) model. Their model is introduced here because of its similarity to the current model and because it will be compared to the current model at several points.

The Miikkulainen and Dyer model sequentially processes the words of a text into propositions, then into schemas, and back out to propositions and then words. The portion of the model of interest here is the portion that combines propositions into a schema. Their model represents propositions as vectors of activation across an input layer, sequentially integrates these propositions into a schema representation, and learns the mapping from propositions to schemas through practice. The architecture is a simple recurrent network (Elman, 1990).

An input proposition is represented as a set of thematic slots such as agent and patient. The concepts that fill these slots have a two-part representation called Content + ID. The content portion is a distributed, micro-feature representation that is learned by the model itself through a procedure called FGREP (Miikkulainen & Dyer, 1989). The content portion comes to encode the meaning of a concept according to its usage in the training corpus. The ID portion is a unique pattern stipulated for each concept. The ID pattern differentiates similar concepts, like Barney and Fred, that might have identical content portions.

A schema is represented as a set of script roles. For example, in a restaurant there would be script roles for the patron, the type of restaurant, and the food ordered. The script roles involved in each context (e.g., restaurant)
are stipulated, but the representations of the concepts that fill the roles is again the learned Content + ID concept representation.

The model is able to perform a number of important knowledge-intensive comprehension processes. That the model does not learn the script roles in each context for itself, however, is a shortcoming that is addressed in the model presented here. Learning is advantageous because it leaves the model unrestricted to any a priori schema representation defined by a programmer. Learning is also advantageous in that the burden of developing a schema representation falls to the model and its training environment, where it ultimately belongs.

The goal of this article is to describe a constraint-satisfaction model of text comprehension similar to the Miikkulainen and Dyer model and tie its processing more closely to a greater set of psychological results concerning (1) representing the explicit text, (2) resolving pronouns, (3) drawing coherence and prediction inferences, (4) revising on-going interpretations as new information becomes available, and (5) sharing knowledge learned in different contexts. Additionally, the model is designed to learn its own schema representation in a hidden layer. This larger learning component in the model should enhance our understanding of what is required from a corpus of experiences, and from the training task posed to the model, to achieve good performance. The model is called the Story Gestalt model because it computes an interpretation for the whole event described in a short text.

Comprehension Tasks

**Representing Multiple Propositions.** A basic requirement for text comprehension is the ability to represent more than one proposition at a time without becoming confused about who did what.

**Resolving Pronouns.** Pronouns create substantial ambiguity in text. Language cues, such as recency and the gender and number of a pronoun provide some constraint on an interpretation (Carpenter & Just, 1977; Corbett & Chang, 1983). Additionally, the situation itself can often provide some constraint (Corbett & Chang, 1983; Hirst & Brill, 1980). The comprehension process should be able to use every available constraint to help compute the referent. 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.
The psychological evidence that readers draw prediction inferences is mixed: Some experiments and methods find evidence (e.g., Graesser, 1981; McKoon & Ratcliff, 1986, 1989), whereas others do not (e.g., Potts, Keenan, & Golding, 1988; Singer, 1990, for a review). A reasonable explanation is that prediction inferences are activated only weakly according to their support from the text. This weak activation can be difficult to detect and therefore it may produce nonsignificant results by some experimental methods.

Coherence inferences, on the other hand, are fully activated and inferred (Potts et al., 1988; Singer, 1990). Following the same line of argument, the strength of coherence inferences results from their stronger support in the text. To simulate this interpretation of the empirical results, models of text comprehension should 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 initial interpretations unlikely. Rumelhart (1981) asked subjects questions about their interpretation as they read the following text.

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. (p. 213)

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, until after reading the final sentence, most subjects believed that the text was about a car salesman.

The idea is that readers update and refine their interpretations as each new piece of information is processed (Carpenter & Just, 1977). Most texts elicit revisions that are not very dramatic (Graesser, 1981). They require only updating the strength of interpretations or altering predictions and very recently processed information. "Garden path" texts, like Rumelhart's, however, can precipitate more dramatic revisions.

Knowledge Sharing and Generalization. A critical question for any system that learns is how it fares on novel examples. If the novel text is composed of a new arrangement of familiar pieces, then a comprehension system should be able to understand it. The difficulty lies in removing the pieces from the contexts in which they were learned, and recombining them appropriately in the new context (see Schank, 1982, for a discussion).

A related issue is the degree to which the novel text violates old script and schematic knowledge. Script violations tend to interfere with accurate comprehension and recall of texts. Graesser, Woll, Kowalski, and Smith (1980)
examined the recall of typical and atypical actions in short texts. Recall of the atypical material was initially good, but deteriorated rapidly. Similarly, Bartlett (1932) found that unusual information in stories becomes regularized over time to fit with established script knowledge.

Bartlett presented British university students with highly unusual Native American folk tales. The students forgot bizarre events, reorganized sequences of actions, and invented information to fit their interpretations. The only details and unusual information that subjects were able to recall were those that played important roles in the subjects' interpretations of the stories.

Given these findings, a model ought to comprehend and recall novel texts, but have difficulty recalling atypical and script-violating information. The aberrant information should tend to be regularized to match the dominant interpretation of the text, or it may be simply forgotten. The Story Gestalt model does not, however, include a mechanism to address the increase in forgetting over time.

**SIMULATION 1**

**Method**

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

**Corpus.** The corpus consists of a large number of texts that describe events in six different contexts: for example, going to the beach or going to a restaurant. The texts are extremely simple: short sequences of actions and descriptions of characters and the situation. Each text is represented in the input to the model as a sequence of propositions.

Each proposition is represented as a predicate and a set of thematic roles. The thematic roles are agent, 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). A predicate is used both for actions and for the predicates of descriptions. For example, the proposition for *it was sunny at the beach*, would be (predicate = weather, location = beach, attribute = sunny).

The concepts are represented locally, so that each possible concept for each thematic role is represented by a separate, single unit. Consequently, there are 20 units to represent each of the 19 agents, plus a unit for and so
### TABLE 1
Concepts in the Corpus

<table>
<thead>
<tr>
<th>Role</th>
<th>Number</th>
<th>Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>agents</strong></td>
<td>20</td>
<td>people, vehicles, policeman, waiter, judge, and.</td>
</tr>
<tr>
<td><strong>predicates</strong></td>
<td>34</td>
<td>decided-to-go, distance, got-in, drove, proceeded, gave, parked, swam, surfed, hung-ten, played, weather, returned, mood, found, met, quality, ate, paid, brought, counted, ordered, served, enjoyed, tipped, took, tripped, made, rubbed, ran, tired, won, threw, sky.</td>
</tr>
<tr>
<td><strong>patients/themes</strong></td>
<td>34</td>
<td>people, vehicles, ticket, volleyball, restaurant, food, bill, change, expensive-wine, cheap-wine, credit-card, drink, pass, slap, cheek, kiss, lipstick, race, trophy, frisbee.</td>
</tr>
<tr>
<td><strong>recipients/destinations</strong></td>
<td>23</td>
<td>people, vehicles, beach, home, airport, gate, restaurant, waiter, park.</td>
</tr>
<tr>
<td><strong>locations</strong></td>
<td>6</td>
<td>beach, airport, restaurant, bar, race, park.</td>
</tr>
<tr>
<td><strong>manners</strong></td>
<td>9</td>
<td>long, short, fast, free, pay, small, not, politely, obnoxiously.</td>
</tr>
<tr>
<td><strong>attributes</strong></td>
<td>10</td>
<td>far, near, sunny, happy, raining, sad, cheap, expensive, clear, cloudy.</td>
</tr>
<tr>
<td><strong>people</strong></td>
<td>12</td>
<td>5 men, 5 women, He, She.</td>
</tr>
<tr>
<td><strong>vehicles</strong></td>
<td>4</td>
<td>jeep, station-wagon, Mercedes, Camaro.</td>
</tr>
</tbody>
</table>

Note. The thematic role name appears in boldface, the number following the role is the number of concepts that can fill that role. Vehicles and people are variables occurring in several roles that take on one of the values shown at the bottom of the table.

that two characters can be agents in a single proposition. There are 34 different units to represent each of the 34 patients, and so on (see Table 1 for a complete list of concepts and roles). If there is no filler for a role in a particular proposition, no unit is activated for that role. All together, 136 units are required to represent all of the propositions. It is important to understand that this localist representation pertains only to the input and output layers. The internal hidden layers are free to develop distributed representations.

The corpus of texts is generated from a set of six scripts, one for each context: activities at the beach, picking someone up at the airport, eating at a restaurant, flirting at a bar, running in a footrace, and playing frisbee in a park. Each script is a tree structure where the nodes are actions or descriptions. Each branch represents a different sequence of events. Generating a text involves probabilistically choosing a path through this branching structure. These scripts were designed to produce a corpus with particular types of variability and regularity for testing the model's processing. No attempt was made to produce natural texts.
A text is generated according to the following procedure. First, one of the scripts is picked at random. Second, a sequence of actions and descriptions is generated by probabilistically choosing a path through the script. For example, within the restaurant script there are paths for a fancy and a cheap restaurant, and subpaths to describe different specific sequences of actions. In fact, there are 20 different restaurant event sequences.

Sometimes a choice must be made whether or not to include a particular action, for example, surfing while at the beach. If surfing were left out of the text, a nil proposition would be created that contained the predicate surfing and a nil for every other role. When the model is questioned about surfing, it should answer using this nil proposition.

For the third step, it becomes useful to distinguish between input texts and the events that they describe. In this step, the event remains the same, but the input text is altered. Specifically, propositions in the text are removed with a probability of .15. This step makes the input texts incomplete and creates the opportunity for the model to draw inferences. The model will still be asked questions about these propositions, and the model must respond with the correct complete proposition.

The fourth step is to instantiate characters' names and car makes with specific values. Finally, in the fifth step, characters' names in the input text may be replaced with pronouns. Each instance of a name in each of the propositions is changed to a pronoun with a probability of .25. Again, no attempt was made here to produce natural sounding texts.

This fifth step, like the third, affects only the input text. The event description retains the characters' names. When questioned, therefore, the model will have to activate the correct character in each case: It is not allowed to answer with pronouns.

Across the six scripts, the total number of different events is 28,480. As noted above, there are 20 different sequences of restaurant events. Each of these involves 2 of 10 possible characters and one of four possible vehicles for a total of 7,200 instantiated restaurant events (20 * 10 * 9 * 4). The number of possible input texts is much higher because of pronouns and missing propositions.

_Architecture and Training Regime._ To comprehend one of these texts, the model processes the propositions one at a time, and evaluates the information in each proposition to build and refine iteratively a representation of the whole text. 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. Each question is a predicate from a proposition in the event and the network is required to complete the whole proposition from the predicate alone.
The model presented here has essentially the same architecture as the St. John and McClelland (1990) sentence-comprehension model. St. John and McClelland's Sentence Gestalt model processes sentence constituents into a propositional representation, whereas the current model processes propositions into a schema representation. That the two models share a common architecture is interesting. The reasons for this similarity are first, that the knowledge-intensive processes simulated by each model can all be viewed productively as constraint satisfaction operating in a PDP network. Second, the architectures of the two models are relatively simple and general. A recurrent part (Part A, in Figure 1) is used to encode propositions sequentially from a text, or to encode sentence constituents sequentially from a sentence, into a gestalt. A second, question-answering part (Part B) is used to train the hidden gestalt layer.

Part A of the network receives each proposition in turn and processes it to update its representation of the text. 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 text at each point during processing. At the beginning of the text, nothing is known, so the units all have 0.0 activation. Activation from the intermediate-combination layer feeds forward to produce a new pattern of activation in the story gestalt layer. The new story gestalt then represents the interpretation as well, because it can be based on the information provided up to that point in the text.

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 text. 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. Bare predicates make good questions because they form unambiguous questions (given that no predicates are repeated), yet completing the propositions they belong to requires the model to fill in a substantial amount of information. Questions about all the propositions in an event, whether explicit or inferred, can be asked by this method.

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 backpropagated (Rumelhart, Hinton, & 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 $j$.

$$C = - \sum_j [T_j \cdot \log_2 (A_j) + (1 - T_j) \cdot \log_2 (1 - A_j)]$$

The model is trained after each proposition on the part of a text 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 third proposition were missing, the model would only be asked about it after the fourth proposition has been presented.

**Results**

The model was trained on 1,000,000 texts generated from the corpus using a learning rate of 0.0005 and a momentum of 0. The learning rate specifies the size of weight changes on each backpropagation cycle, and the momentum
specifies the proportion of weight change from the previous cycle that is carried forward to the current cycle.

The model performs correctly on the majority of texts in the corpus after 350,000 training texts. At this point in training, as discussed later, however, pronoun resolution is not always handled correctly. Also, the reliability of information, or cues, in the texts has not been learned accurately: More reliable cues do not always outweigh less reliable cues. Past this point, the model slowly learns the difficult pronoun resolution cases and learns to weigh conflicting cues more accurately. The extent of training needed to eke out the last bits of information in the corpus is somewhat worrisome. This condition might be ameliorated by using special training procedures that allow the model to focus its training on problem cases, for example, by training problematic texts more frequently. The primary concern of the current research, however, was the ultimate performance of the model, so this possibility for speeding up training was not explored.

*Pronoun Resolution.* The model's pronoun-resolution abilities can be tested by presenting the model with texts containing propositions with pronouns. The model is then questioned with the predicates from these propositions. The completed propositions produced by the model can then be checked to see if the correct characters are activated. In cases where the text is ambiguous, the completed propositions can be checked to determine if all of the plausible characters are partially activated.

The two pronouns tested were "he" and "she." Other proper pronouns were not assessed, and pronouns like "it" that can refer to whole events are beyond the scope of the model. When "it" refers to a whole event, that event becomes a single role in another proposition. For example, in "Wilma broke her leg skiing. It kept her out of commission for weeks," the "it" in the second sentence refers to the whole proposition in the first sentence. The problem is that the propositional representation used in the model does not allow embedded propositions.

A number of cues for pronoun resolution are available in the corpus. The simplest occurs when there is only one character in a text. In this case the pronoun should be constrained to be that character. The model handles this simple case correctly. If the character is initially unknown, the model activates each character slightly. When the text eventually specifies the character, the model immediately resolves each past pronoun fully.

Correct resolution is more difficult when two characters are involved in a text because each must be assigned to the correct roles and remembered without error. For example, the restaurant script contains two characters, yet the model recalls each character correctly. One constraint the model learns to help perform this task is a script-role constraint that occurs when one character performs a set of actions. For example, the character who pays the bill is also the character who orders and the character who receives
change from the waiter. If the model knows the character from one of the propositions, it correctly resolves a pronoun in any other.

The park script is specifically designed to test three other pronoun-resolution cues: gender, recency, and the situation. The park script contains a pronoun in the agent role of its final proposition, and it varies the availability and reliability of these three clues to the referent of that pronoun. The model should learn to use each cue, when available, to the degree it is reliable.

**Example of a Park Text**
Andrew and Roxanne decided to go to the park.  
The weather was sunny.  
**recency cue:** Andrew ran through the park.  
**gender and situation cues:** He threw the frisbee to Roxanne.

The gender cue, coded by the pronoun itself, is only useful when the characters in the text differ in their genders. When this condition holds, the gender cue is completely reliable.

The recency cue says that the agent of the previous sentence is likely to continue to be the agent of subsequent sentences. In the corpus, this cue was reliable in 80% of the cases. In the remaining 20% of the cases, the other character was the referent of the pronoun. Recency is unavailable when the second-to-last sentence is missing from the text.

The situation cue says that the recipient of the frisbee cannot be the agent, thereby ruling out one of the characters. The situation cue is always reliable when available, but in half of the training texts, the cue was removed by replacing the final sentence with “S/He threw the frisbee at a tree.”

The gender and recency cues are learned rapidly, but the situation cue proved difficult to learn. Though eventually learned correctly, the model’s initial performance is interesting: The presence of the alternative character in the recipient role produced activation for that character in both the recipient role and in the agent role. This effect can be described as poor role binding. If the recipient character is poorly bound to his role he may be partially bound to both the recipient and the agent roles. The result is partial activation of both characters in both thematic roles. This effect suggests that the model has not completely learned a representation that binds variables, in this case characters, to their respective roles.

The recency cue appears to work similarly. The presence of the recency cue tends to raise the activation across the boards of the character it predicts. In fact, another proposition, `<Person>` drove a `<vehicle>` to the park, which can be included before the pronoun, also has an effect on the resolution of the agent: The proposition increases the activation for the pronoun of whichever person drove. This increase occurs despite the fact that in the corpus, the driver is a completely unreliable cue to who threw the frisbee.

This poor role binding eventually disappears given enough training. In pursuit of good role binding, the model was retrained on just the park script
Table 2 shows the model's performance on the park corpus. The activations of the correct and incorrect referents of the agent role in the final sentence are shown for each combination of cues. The gender and situation cues can be either present or absent, and the recency cue can predict the correct referent, the incorrect referent, or it can be absent. The activations shown are averages from four test cases per cell. Activations range between 0.0 and 0.9. The correct activations, as defined by the reliabilities of cues in the corpus, are shown in parentheses. Note that the cues may sometimes conflict. For example, gender can cue one referent and recency can cue the other. The model resolves the pronoun in accord with the more reliable cue, gender.

Two previous models of pronoun resolution examined similar types of cues. Allen (1987) trained a distributed three-layer network to resolve pronouns based on gender or a semantic cue. Elman (1988) trained a recurrent distributed network to resolve pronouns based on a syntactic, structural cue called c-command (Reinhart, 1983). What is interesting about Elman's model is that c-command is defined structurally in terms of abstract constituents, like NP and S, and tree structures, yet the model was able to perform the pronoun resolution correctly without requiring the explicit use of such structures. The similarity of the Story Gestalt's architecture to Allen's architecture and especially Elman's architecture suggests that they should all produce similar performances.

Inference. The model's propositional inference performance can be examined by presenting to the model minimal texts that contain few proposi-
TABLE 3

Inference

Input Text
Albert and Clement decided to go to a restaurant.
The restaurant was expensive.
Clement paid the bill.

<table>
<thead>
<tr>
<th>DECIDED TO GO</th>
<th>Questions</th>
<th>ORDERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent:</td>
<td>Clement</td>
<td>agent:</td>
</tr>
<tr>
<td></td>
<td>.9</td>
<td></td>
</tr>
<tr>
<td>Albert</td>
<td>.4</td>
<td>Albert</td>
</tr>
<tr>
<td>and</td>
<td>.9b</td>
<td></td>
</tr>
<tr>
<td>destination:</td>
<td>restaurant</td>
<td>patient:</td>
</tr>
<tr>
<td></td>
<td>.9</td>
<td></td>
</tr>
<tr>
<td>QUALITY</td>
<td>patient:</td>
<td>expen. wine</td>
</tr>
<tr>
<td>restaurant</td>
<td>.9</td>
<td></td>
</tr>
<tr>
<td>value:</td>
<td>expensive</td>
<td>.9</td>
</tr>
<tr>
<td>DISTANCE</td>
<td>restaurant</td>
<td>patient:</td>
</tr>
<tr>
<td>restaurant</td>
<td>.9</td>
<td></td>
</tr>
<tr>
<td>value:</td>
<td>far</td>
<td>.9</td>
</tr>
<tr>
<td>TIPPED</td>
<td>agent:</td>
<td>patient:</td>
</tr>
<tr>
<td></td>
<td>Clement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Albert</td>
<td></td>
</tr>
<tr>
<td>manner:</td>
<td>small</td>
<td>.4</td>
</tr>
<tr>
<td></td>
<td>big</td>
<td>.1</td>
</tr>
<tr>
<td></td>
<td>not</td>
<td>.0</td>
</tr>
</tbody>
</table>

a The model is shown a short restaurant text, then questioned about a number of propositions from the whole event. The numbers indicate the activation, from 0 to .9, of active concepts.

b In the decided to go proposition, there are two agents, as indicated by the and unit. In every other role and in every other proposition, there should be only one agent. Without the and, activation of multiple concepts within a role represents alternative fillers of that role.

The model can draw inferences based on a variety of information. The model uses previous experience with Clement from the training corpus to infer that he will order cheap wine. The model uses previous experience to
the effect that expensive restaurants are always far away to infer that the distance to this restaurant is far. The model also uses script-role information that Clement paid the bill to infer that the waiter will return the credit card to him rather than Albert, that he must have ordered, and that he will leave the tip.

These inferences are typically activated to saturation (.9 on a scale of 0-.9). The explanation is that many of the correlations in the training corpus were absolute: all expensive restaurants are far away, and the payer of the bill is always returned the credit card.

The model can also draw more complex inferences. The tip size depends upon three factors: who tips (based on who paid), whether that character is cheap or extravagant (based on the previous experience that cheap characters tip small), and the quality of the restaurant (because, in the corpus, tips were never left in cheap restaurants). The model had to combine all three factors to determine the tip size. It performs correctly in each case.

Default information based on base-rates can also be inferred. The beach script provides a good example: In 80% of the texts the weather is sunny, but in 20% of the texts the weather is raining. Sunny, then, is the default, and when no weather is specified in a text the model should activate sunny .8 and raining .2. The model, on average, activates sunny .7 and raining .2.

Finally, Table 3 shows the model’s ability to draw both prediction and coherence inferences. Coherence inferences concern propositions that lie between explicit text propositions. Prediction inferences concern propositions that lie after the final explicit text proposition. For example, ordering is a coherence inference because it occurs before the explicit proposition about paying. Tipping is a prediction inference because it occurs after paying. In the Story Gestalt model, all inferences are processed the same way. Any information correlated with the explicit text is activated immediately as the text is read.

Though the locations of inferences does not matter, the reliability of the available cues in a text does matter. The bar script provides examples: Early propositions in the script predict the final proposition with varying reliability. When no cues or only weakly reliable cues are present, the model activates each prediction inference to a degree. When more reliable information is available, the model’s predictions are correspondingly stronger. The following section on revision demonstrates this effect and shows how the model adjusts its predictions on-line as more information becomes available.

*Revision.* The bar script tested the model’s ability to utilize the reliabilities of cues and to revise its predictions as more information becomes available. In these texts, two intermediate propositions predict, with different strengths, the final proposition of the text. After each proposition is presented, the model’s prediction for the final proposition can be observed to
see how it changes based on the cumulative information from the proposi-
tions presented so far.

Organization of the Bar Script

<Person1> decided to go to the bar.
<Person1> made a polite/obnoxious pass at <person2>.
<Person2> gave kiss/slap to <person1>.
<Person1> rubbed lipstick/cheek.

After just the first proposition is presented, rubbed lipstick and rubbed 
cheek are equally likely, and the model should activate both possibilities 
partially. The second proposition predicts the ending probabilistically. A 
polite pass predicts rubbed lipstick 70% and rubbed cheek 30%, while an 
obnoxious pass predicts rubbed lipstick 30% and rubbed cheek 70%.

Table 4 shows the model’s predictions of the ending given texts contain-
ing different cues. The activations shown are averages from three test cases 
per cell. The correct activation of each ending, as defined by the reliabilities, 
is shown in parentheses.

The model’s activations correspond well with the probabilities. With no 
cues, the model activates both endings partially. Once the second proposi-
tion has been presented, the model again predicts both endings, but revises 
its predictions to make the more likely ending more active.
After the third proposition, the model again revises its predictions. Because the third proposition is completely reliable, the model revises its predictions to fit with that cue. The most interesting cases are when the model has been leaning in one direction, but then gets a stronger cue in the opposite direction, for example, obnoxious pass, which predicts rubbed cheek, and then kiss, which predicts rubbed lipstick. Even in these more serious revision cases, the model performs correctly.

A final note of interest is that earlier in training, the model had not learned the correct reliability of each cue. After the third proposition had been presented, the second continued to influence predictions. With further training, the model learned to ignore the weaker cue when the completely reliable cue was present.

Knowledge Sharing and Generalization. How well does the model's learning generalize to novel texts? Can the model correctly process and recall novel texts composed of familiar pieces? An important variable is whether the novel pieces are individual concepts or whole propositions. The architecture of the model makes this variable important because the incorrect concepts within a proposition are trained to have zero activation whereas incorrect and irrelevant propositions are simply never trained.

First, the recall of texts with novel concepts will be tested. This is followed by some theoretical discussion of generalization, and then by the recall tests of texts with novel propositions. To test recall of novel concepts, texts were created in the standard fashion, but a script-violating concept replaced one of the standard concepts, for example, replacing food with frisbee. In all cases when questioned, the model completely activated the standard concept rather than the actually presented, script-violating concept.

Additional cases were tested in the following way. During training some characters and vehicles were restricted from being seen in particular contexts. Each was seen, however, in several other contexts. At test time, new texts were created that included these previously restricted concepts, for instance, driving a Camaro to the beach and Andrew visiting a bar. In nearly every test case, the previously restricted concept received no activation, and activation was evenly distributed across the trained concepts—as if the model were saying, "I don't know which car was driven to the beach, but I'm sure it wasn't the Camaro."

To some degree, this effect is reasonable. People have been found to have a similar tendency to misread words when the sentence context is particularly strong. For example, subjects who read the question, "How many animals of each kind did Moses take on the ark" responded "two," often without noticing that Moses is not the correct Biblical figure. Erickson and Mattson (1981) termed this phenomenon a "semantic illusion."
People can, however, read more carefully and reduce the likelihood they will fall for illusions (Reder & Cleeremans, 1990). Rather than merely ignoring the actual input, then, it would be better if the model demonstrated some competition between the word meaning and the context, but allowed the word meaning to dominate. The actual word meaning and the context-supported meaning would both be partially activated, but the actual word meaning would be activated more strongly. Semantic illusions would occur when the input was not read carefully.

I now present a theory to explain these poor generalization results. This theory has important ramifications for training networks to generalize in many tasks. During training, the model learns two types of constraints or mappings between the input and the output: identity constraints and associative constraints. Identity constraints encode the auto-correlation of a concept with itself: Camaro means Camaro, frisbee means frisbee, and ate means ate. Identity constraints produce veridical recall and underlie the ability to understand novel texts by understanding the individual contributions of each familiar part.

Associative constraints encode the correlations between associated concepts: ate is associated with food, restaurants, paying checks, and so on. Associative constraints underlie the ability to draw inferences to fill in and anticipate information. They work against generalization because each familiar part of a novel text brings along conflicting sets of associations. The associative constraints from the restaurant context actively inhibit frisbees as food.

St. John and McClelland (1990) found one way to improve the processing of novel inputs. In their sentence-comprehension model, they found that a highly combinatorial corpus, in which each word appeared in many different contexts, produced good comprehension of novel sentences. Why? Essentially, a combinatorial corpus removes the correlations that produce the associative constraints; only the identity or auto-correlations remain. Perhaps a highly combinatorial corpus will also improve the Story Gestalt model's abilities to process novel texts. This hypothesis is explored in Simulation 2.

Continuing with the script-violation experiments, the model's ability to recall texts with novel combinations of propositions was also tested. Three of the scripts, beach, restaurant, and park, contain the driving script within them. Fifty percent of the time, only the single proposition, \(<\text{person}>\text{ drove to } \langle\text{destination}\rangle\) is included, but the other 50% of the time, the entire set of driving propositions is included. The airport script, however, contains only the single proposition, \(<\text{Person}>\text{ drove }\langle\text{vehicle}\rangle \text{ to airport for a long time}\). Proposition sharing was tested by presenting the model with a new airport text that included the entire set of driving propositions.
Organization of the Driving script

The distance to the <destination> was near/far.
<Person> got in <vehicle>.
<Person> drove <vehicle> to <destination> for a short/long time.
<Person> proceed to <destination> fast.
<Person> parked at <location> for free/pay.

The model strongly activates the driving propositions. The model can both recall them correctly as presented or predict them correctly from the initial proposition, <person> decided go to to the airport. The model does have difficulty, however, modifying the driving propositions to fit the airport context: The model tends to activate the restaurant or beach as the destination and location, rather than the airport. Within the driving script there is a bias to turn off airport as the location. In processing the test texts, the identity constraint for airport as a location in these propositions is not strong enough to overcome this bias.

Another text described a trip to the beach and contained the irrelevant proposition Andrew ran to the gate at the airport. When questioned with ran, the model activates the full proposition. Despite the fact that this text is incoherent, the model does not complain by, say, activating the answer to this question only weakly.

The question-answering performance on these novel texts arises from the architecture of the model. During training, the model is only questioned about relevant, appropriate propositions. It does not learn which propositions cannot go together. This procedure works to the advantage of legitimate novel texts. The model has little trouble recalling the novel combinations of propositions. This procedure, however, rings false for irrelevant propositions. The model ought to balk at irrelevant propositions. I will return to this issue in the General Discussion.

Discussion

The model demonstrates a number of successes. It resolves pronouns accurately. It combines multiple sources of information, or cues, and allows the stronger more reliable cues to dominate. The model also strongly draws both coherence and prediction inferences according to their likelihoods. Finally, the model revises its interpretations appropriately as more information is processed.

The main shortcoming of the model's performance is its difficulty in sharing information between contexts and accurately recalling script-violating information. Weak identity constraints and overly strong associative constraints lie at the root of this problem. Simulation 2 examines this issue. In particular, it examines the impact of the corpus on knowledge sharing and generalization.
SIMULATION 2

Script Violations
To address knowledge sharing and generalization, consider first the credit card example. The model is trained with two basic situations: cheap restaurants and expensive restaurants. Paying a check with cash is only paired with cheap restaurants, and paying with credit-cards is only paired with expensive restaurants. The model learns an ironclad rule that cash goes with cheap restaurants and credit-cards go with expensive restaurants. The model has no knowledge or experience that any other situations are possible.

What is needed is some understanding that the alternate situations can occur. How might such knowledge be acquired? An interesting possibility is to add additional texts to the corpus that break the cash-cheap/credit-card-expensive rule in other contexts. These texts would also be about buying things, but they would occur in nonrestaurant locations and they would not contain the regularities between expensive and credit-cards or cheap and cash.

In identity and associative constraint terms, to process these texts correctly, the model would have to learn the identity constraints for each concept. The model would not be able to rely on the associative constraints between concepts to answer questions because there are no regularities from which to learn associative constraints. The identity constraints that are learned might transfer to the restaurant context so that paying with a credit card in a cheap restaurant could be represented and recalled correctly.

This possibility for improving the model's processing of script-violating texts was tested in a new simulation with a corpus pared down to the essential details. The first corpus contained only restaurant texts. This corpus was designed to produce weak identity constraints and demonstrate the difficulty of representing script violations. The second corpus contained restaurant texts and texts in other contexts designed to improve the learning of the identity constraints.

Organization of the New Restaurant Script

The quality of the restaurant was {cheap, expensive}
The distance to the restaurant was {near, far}
Andrew paid with {cash, credit-card}
Andrew tipped {small, large}

In each corpus, texts were four propositions long. Each proposition contained a role that could take on 1 of 2 values. With four propositions and 2 values for a role in each, there are 16 possible combinations of role values. In the restaurant context, only 2 of the combinations appear in the corpus: The other combinations are never trained. These 2 combinations correspond to the cheap restaurant where the role values are (cheap, near,
The correlations among role values in the restaurant, however, were also learned and are also applied during processing. These associative constraints account for the slight activation of the script-appropriate values. More importantly, if the value of a role in the restaurant context is left out, the model infers the script-appropriate 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. Again we find that the model combines information from a number of constraints. Each has its influence on the activation values. What the model creates, then, is something like a "graded slot."
Of course, sharing knowledge runs in both directions. While the restaurant context gains the use of the identity constraints, the other contexts gain the associative constraints among role values. When a value is missing in one of these contexts, 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.

A similar situation occurs when familiar concepts are placed in novel contexts such as driving a Camaro to the beach. The model was trained on a new corpus that consisted of texts about driving to a location. Each text consisted of the single proposition (agent = Andrew, action = drove, patient = vehicle, destination = location). There were four different possible vehicles and five different possible locations. When the location was the beach, the vehicle was restricted to not be the Camaro.

Once trained on this corpus, the model was tested with the Camaro being driven to the beach. Questioned with the predicate drove, the model responded by activating Camaro 0.0 and the three other vehicles roughly 0.3. Therefore, the four contexts in which the Camaro was used were not sufficient in strengthening the Camaro identity constraints to overcome the associative constraints from beach.

A fresh model was then trained on a new corpus that expanded the total number of locations to 10. The number of vehicles was also increased from 4 to 10. In this corpus, the Camaro is driven to 9 of the 10 locations, but not driven to the 10th location, the beach. Again, the model was trained and then tested with the Camaro-beach text. The Camaro unit was activated .5 while alternative vehicles were kept at 0. For comparison, vehicles actually trained in the beach context are typically activated to .9. The correct activation of the Camaro unit, therefore, is moderated by an associative constraint from beach.

What is going on in these corpora? Why does the first restaurant corpus lead to associative constraints, and the third to identity constraints? Associative constraints are easier to learn because the input needs to be processed less. The question itself can be used to answer many questions, and if any part of the input is processed, associative constraints can be used to fill in the remainder.

It is only when the corpus is strongly combinatorial that identity constraints become well learned. When different parts of texts are uncorrelated, each part must be processed individually to produce the correct results. In the nonrestaurant contexts, the regularity between cheap and cash does not exist; the pairing of cheap and cash is only as likely as the pairing of cheap and credit-card. The model has to process both propositions to be able to answer questions about each. Under this condition, the identity constraints were learned, and novel texts could then be represented correctly.

An interesting comparison can be made with Miikkulainen and Dyers's (1991) Content + ID method of concept generalization. As mentioned
previously, concepts have a two-part representation. The content part specifies their basic meaning and the ID part identifies each individual within a conceptual class.

During training, the ID portion of concepts is constantly varied. The model learns to "pass through" the ID portion from input to output. Meanwhile, the model learns to use the content portion of concepts to compute the text meaning and draw inferences. The model is learning strong identity constraints to represent and recall the ID portion of concepts while it learns associative constraints to draw inferences from the content portion.

Generalization to new instances within a concept class is easy because the identity constraints insure the faithful transmission of the ID portion. In fact, the method allows entirely novel instances of a class to be processed correctly because the content portion is familiar, and the identity constraints transmit the ID portion. Changing the concept class, on the other hand, should not produce good generalization because the content portion is much less variable: Strong associative constraints have been learned.

**Real Violations**

In each of the cases so far, the novel text was sensible, and the problem was getting the model to override a regularity it had learned from the corpus. It is easy, though, to produce novel texts that are not sensible. These "real" semantic violations will constitute fairly atypical activities and concepts. In accord with Bartlett's (1932) folktale study, these very atypical activities should be difficult to represent and recall. The example used here is that *semis* (tractor trailer) rarely, if ever, go to the beach. Critically, there is a good reason for this: Beaches are for recreation and semis are used only for commerce. Because there is a reason that we have not seen semis at the beach, we should be skeptical when we hear a text about one. This novel text should not be understood and recalled as blithely as a novel text about a *Camaro* going to the beach.

Actually, bizarre events can be recalled by subjects under certain circumstances, if the delay between reading and recall is short. An important factor is that subjects spend extra time reading the bizarre sentences. It seems that this extra time produces better encodings of these sentences (Bellezza, 1983), though they still decay rapidly. The Story Gestalt model has no mechanism for increasing reading time, so it corresponds to a situation where subjects are forced to continue reading without opportunity to pause.

To capture the idea of real violations, another corpus was created that contained 10 commercial and 10 recreational locations. Each text contained two propositions. The first specified whether it was a commercial or recreational location, and the second proposition was the proposition about driving to a location used previously. In the commercial contexts, each of 10 vehicles could be used. In the recreational contexts, semi was never
used, and in the beach context, both semi and Camaro were never used. Camaro, though, was seen in each of the other recreational contexts and in each of the commercial contexts.

The quality of the <location> was {recreational, commercial}.

Andrew drove a <vehicle> to the <location>.

After training, the model was tested with texts involving different vehicles in the beach context. Actually trained vehicles were activated .9. In the novel text containing Camaro, Camaro was activated .6, and in the novel text containing semi, semi was activated .1. Semi was, however, the most active unit: In all cases, the other vehicle units were 0. As predicted, then, the "real" violation disrupts recall more than a "sensible" violation.

The model's sensible performance derives from the additional associative regularities involving the first proposition. The attribute recreational is never paired with semi, so the model learns an associative regularity that semis are never seen in recreational contexts. Meanwhile, the model learns two associations with beach: no semis and no Camaros. The activation of Camaro in the beach context is reduced by the constraint from beach. The activation of semi is reduced more strongly by the constraints from both beach and recreational. It is important to understand that there is still no logical, all-or-none rule that explains why semis are never driven to the beach. There is simply an additional, strong, correlation against it.

It may be the case, however, that an explanation could be modeled as a matrix of correlations that strongly blocks generalizations. Semis conjure up associations of commerce and industry far removed from beaches. Perhaps a more sophisticated corpus, which contained this explanatory information, and training questions such as why? and why not? could produce explanations as answers.

Scaling to Larger, More Complex Corpora

Considerations of how the model will scale to larger corpora or more complex texts give further insight to the model's capabilities. One question is simply, how will the model fare as the number of texts in the corpus is enlarged? How much of the corpus will have to be trained, and how much can be left to generalization? The script-violation studies put us in a better position to understand this question.

A good way to achieve strong generalization is to keep regularities in the corpus to a minimum so that strong identity constraints are learned. The problem is that removing texts to create a generalization set creates regularities, which opens the door to associative constraints.

The solution is to use the combinatorics in a corpus to advantage. If a corpus is sufficiently combinatoric, a large percentage can be removed without creating strong associative constraints. To pursue this idea, a corpus
was created consisting of 20 characters visiting 20 different professionals. Each text was four propositions long. Only the professional’s office and the name of the visiting character varied among texts. Ten characters in each office context were chosen at random to be excluded from the training corpus. Consequently, the model was trained only on 200 of the 400 texts in the complete corpus.

After training, the model was tested for its ability to process and recall the novel, untrained texts correctly. Ten novel texts were picked at random for testing. Each was processed correctly: The correct units were either the only units active, or were the most active units in each case.

In conclusion, omissions from the training corpus produce regularities that will be encoded as associative constraints. These constraints can limit generalization unless strong identity constraints are otherwise learned. Combinatorial corpora, where the texts in the corpus consist of combinations of a set of elements, are ideal for fulfilling this requirement. Each element is seen and practiced in many different contexts, so that its identity constraints are strongly learned.

A second scaling question is how will the model fair on longer texts? How much will training time increase for texts with more propositions? In the first simulation, the number of propositions was not the critical factor on training time. Additional propositions should be quickly learned, and if they are correlated with previous propositions, then little extra storage space in the gestalt layer is needed to retain the propositions for recall.

The critical factor on training time was, instead, complexity. Developing a representation to handle role binding proved to be difficult for the model. Role binding requires something like conjunctions to bind characters to their roles in the texts, and conjunctions are slow to develop.

Besides role binding, complexity can be created by giving texts a hierarchical structure. The propositional representation used in the input and output layer of the model restricts the model from representing hierarchical structures. This issue is discussed further later.

GENERAL DISCUSSION

To begin, I discuss a number of design issues related to the model’s performance and adequacy. Next, I compare the model’s design and performance to previous models. Last, I conclude.

Design Issues

Propositional Representation. The propositional representation, although reasonably convenient, imposes some important limitations on the texts. First, each text can use a particular predicate only once. Because the predicates are used for questions in training the model, if two propositions had
the same predicate, the output for that question would be a blur of both propositions. This situation has to be avoided. One solution would be to use conjunctive questions, such as the agent and the predicate. Placing more information in the question, however, leaves less information to be computed from the text.

Second, the division of propositions into seven thematic roles, such as agent and patient, is arbitrary and is useful primarily for simple actions. Ideally, the model would learn its own representation for propositions just as it learns its own representation for story gestalts.

Similarly, the use of one unit to represent each concept in each applicable role in the input and output is inefficient and will not scale well. Again, ideally, the model would learn these representations in some more compact, distributed form. Miikkulainen and Dyer (1991) demonstrated one method for achieving this result for the content portion of their concepts.

Finally, each thematic role in a proposition can only be filled by a concept rather than an entire proposition. There is no way to represent arbitrarily embedded structures in the output layer. Pollock (1988) described an abstract way to encode and decode embedded structures by recursively unpacking a gestalt-like pattern into its constituents. In his model, the hidden layer of a three-layer network is used as the gestalt, and the output layer represents the constituents. If a constituent is not a terminal, it is copied back to the hidden layer and run through the network again until all the constituents are terminals. Though this procedure suggests that representing embedded structures in a distributed network is possible, this procedure would add considerable complexity to the learning task of the model.

Network Architecture. The architecture is basically a simple recurrent network (Elman, 1990; St. John & McClelland, 1990). The extra hidden layer between the input and the Story Gestalt layer allows complex mappings from input to Gestalt to be computed. The hidden layer between the Story Gestalt and the output implements the question-answering function. A consequence of these extra layers is that the network is deep: four layers of weights lie between the input and the output. Learning becomes slow because the error is distributed among all the weights and reduced as it is passed back from the output toward the input through each layer. The modification of weights close to the input, then, is very slow and requires considerable training. A more general point is that each weight is a parameter that must be set to achieve good performance. With so many weights, credit assignment is difficult. Consequently, training time is long.

Training time is also increased by making the story gestalt layer a hidden layer, thereby forcing the model to learn its own story representation. This situation may be the price paid for increasing the learning component of a model. More sophisticated learning procedures, however, may ameliorate this problem.
A second effect of the network architecture is that the training of incorrect concepts is treated differently than the training of irrelevant propositions. When a proposition is trained, incorrect concepts within the proposition are explicitly trained to have zero activation in the output layer. For example, if Camaro were the vehicle in a text, all other vehicles would be trained to have zero activations. This training produces negative associative constraints between that context and the incorrect vehicles. These constraints account for some of the difficulty in knowledge sharing found in the first simulation. They do not account for all of the difficulty, however, because many other associative constraints between cues may also be learned.

Irrelevant propositions, on the other hand, are simply never trained. Consequently, negative associative constraints between a context and irrelevant propositions are never learned. Recalling script-violating propositions, therefore, is easier than recalling script-violating concepts. How best to resolve this asymmetry is not clear. One possibility would be to train the model to respond irrelevant to a small number of truly irrelevant questions in the course of training each text.

**Corpus.** The characteristics of the corpus can have important consequences on the performance of the model. As we have seen, regularities in the corpus are learned and encoded as associative constraints. These constraints allow the model to draw inferences, resolve pronouns, and revise interpretations, but they limit the ability of the model to represent and recall the atypical activities and concepts in novel texts.

When the corpus is more combinatorial, so that concepts and propositions are seen in many different contexts, the processing of novel texts is much better. Under this condition, the model is forced to learn strong identity constraints. The identity constraints can then be used to process novel texts more successfully.

**Correlation-Based Knowledge.** One further issue is that the model has no intrinsic understanding of concepts like causality. Its knowledge is entirely based on what actions tend to cooccur in its training corpus. People certainly have knowledge of causation, but the specific question here is whether causal knowledge plays any substantive role in text comprehension.

Van den Broek (1990) argued that it does because coherence inferences are drawn in an effort to keep the text causally coherent. During reading, the comprehension system evaluates each sentence of a text to see whether it is necessary and sufficient to explain the chain of events in the text, and draws inferences when it is not. Many experiments (for reviews, see Singer, 1990; Van den Broek, 1990) find that the chain of causal events running through a text is especially well recalled. The explicit evaluation of causality and subsequent drawing of inferences provide a rationale for this effect.
The view advocated here, on the other hand, is that the comprehension system automatically activates all information well correlated with the text. Among this information, lie the causal antecedents of events in the text. The high rate of recall of events on the causal chain results from the especially high correlations between causes and effects, rather than explicit computation or the logical status of causality.

Comparisons Among Algorithms
A large number of models of knowledge-intensive comprehension processes have been proposed. Together, they span a great variety of architectures and model a variety of comprehension processes. The Story Gestalt model represents an improvement in representation and processing in several regards.

A major problem in text comprehension is finding information that should be inferred to make a text interpretation coherent. The search for these inferences is often underconstrained, finding quantities of irrelevant and incoherent information as well as sometimes not finding the relevant information. Charniak (1983, p. 179) provided the example of “John wanted to commit suicide. He went and found a rope.” Models that search for inferences only backward from the final sentence (e.g., Kintsch, 1988) will find quantities of irrelevant information about ropes and may miss the one relevant datum about making a noose. A better search procedure is to search based on all of the available information.

Charniak (1983) proposed this sort of parallel search through a semantic network. As a new sentence is processed, markers spread out from the current sentence and the previous context. When the markers meet, their paths are taken as a set of inferences that connect the current sentence with the previous context. Although this parallel search procedure is far more likely to find connecting paths than a backward search from the current sentence alone, it runs into another problem: Many connecting paths are likely to be found, and most will be entirely senseless. For example, given “Bill wanted to commit suicide. A rope fell down on Fred,” the model will find the path Bill-suicide-kill-hang-noose-rope. The problem is that the spreading markers do not encode anything about the roles concepts play or even which sentences they come from. Charniak’s model has to employ a time-consuming secondary process to evaluate each path and remove the illegitimate ones.

Graded constraint satisfaction (Rumelhart, Smolensky et al., 1986), on the other hand, is a parallel search that converges on an interpretation that best satisfies the constraints found in the text. Irrelevant information is suppressed, but appropriate connecting information is inferred because it is a part of the whole interpretation that the constraint-satisfaction search finds. A number of models (Lange & Dyer, 1989; Miikkulainen & Dyer, 1991; Shastri, 1988) use constraint satisfaction to good effect.
A second problem for some models is that processing the explicit text, resolving pronouns, and drawing inferences work separately. In Kintsch's (1988) model, each possible pronoun assignment is computed, the inferences are added, and then the incorrect assignments are pruned at the same time incorrect inferences are pruned. In Schank and Abelson's (1977) and Cullingford's (1978) Script Applier Mechanism, propositions containing pronouns are first matched to a script, then inferences are drawn, and then pronouns are resolved. The question is why these processes should be separated even though they rely on the same information and ought to be mutually supportive. In the Story Gestalt model, resolving pronouns and drawing inferences are part of the same constraint-satisfaction process and happen simultaneously.

A third problem is that the long-term knowledge, the text interpretation, or both, is often represented in an all-or-none fashion that does not reflect the gradedness of information in the real world (Charniak, 1983; Schank & Abelson, 1977; Shastri, 1988). For example, the likelihood of a causal relationship cannot be represented in some models, yet differences in likelihood are common, and produce differences in reading speed (Keenan, Baillet, & Brown, 1984).

In the Story Gestalt model, as well as in several other models (e.g., Kintsch, 1988; Lange & Dyer, 1989; Miikkulainen & Dyer, 1991), constraints are graded. The likelihood of a relationship can be represented by the strength of a constraint. This representation allows the model to use less than completely reliable information to compute interpretations.

A related problem is that revision and refinement of an interpretation is difficult in models with all-or-none representations because changes to an all-or-none representation are necessarily drastic. Furthermore, incompatible alternative predictions may follow from a text, and it is unclear how to represent them in an all-or-none fashion. Imagine three alternatives, only one of which is very likely. If likelihood were not encoded at all, then all three alternatives would be equally likely by default, and it is not clear which should be predicted. If likelihood information were encoded but predictions were still made all-or-none, then it is still unclear which predictions should be drawn. Consequently, some systems (Charniak, 1983; Schank & Abelson, 1977; Shastri, 1988) are conservative and rarely make predictions or other risky inferences that might require revision. Graded constraints and graded activations allow alternative predicted information to be activated to the degree it is supported by the text.

With regard to generalization, sharing information learned in different contexts is a well-known problem (Schank, 1981). Knowledge in Schank and Abelson's (1977) model is tightly bound to individual scripts making it difficult to share. Schank's (1982) later work on MOPs addresses this problem by breaking up scripts into a hierarchy of much smaller pieces. For
example, visiting a dentist involves very general MOPs encoding knowledge about exchanges and services, as well as specific MOPs encoding knowledge about a particular dentist. The knowledge contained in the more general MOPs is then easier to apply to novel situations. Semantic networks provide another method for encoding knowledge as separate pieces: as individual propositions. But knowledge sharing remains a problem because it is difficult to know which knowledge is specific and which is general enough to transfer.

Constraint-satisfaction models encode knowledge as separate pieces via the identity constraints. Novel texts, then, are simply novel combinations of constraints. The difficulty is that associative constraints, acting like tightly bound script knowledge, can interfere. Qualities of the learning environment, such as the corpus, affect the relative strength of these constraints and thereby affect knowledge sharing. The Story Gestalt model demonstrates these ideas. Clearly, though, these ideas will have to be pushed and expanded greatly if they are to match the level of sophisticated processing exhibited by MOPs.

Finally, the acquisition and organization of knowledge is a difficult and typically unresolved issue in most comprehension models. In particular, all-or-none representations make learning difficult because decisions such as whether to include information in a script, when to start a new script, or when to use a variable must be decided all-or-none (Rumelhart, Smolensky et al., 1986; Schank, 1981). Learning in graded-constraint representations demonstrated here, and by Miikkulainen and Dyer (1991), may alleviate this problem by encoding the likelihood and strength of relations.

Of course there are criteria on which the Story Gestalt model does not fare as well as other models. Schank and Abelson’s (1977) and Cullingford’s (1978) SAM model is able to represent complex, hierarchical texts and give higher status to important actions. A number of models (including SAM; Lange & Dyer, 1989; Wilensky, 1983) can also represent and reason about the goal structure of texts. Among connectionists, a number of researchers have chosen to work with more localist models because it is better understood at this time how to represent hierarchies in localist models (Dyer, 1991).

The Story Gestalt model can perform none of these tasks: It does not represent goals and is only able to represent simple, nonhierarchical texts. The application of graded constraint-satisfaction models like the Story Gestalt model to these tasks awaits further research into better representations and architectures. But the positive qualities of the Story Gestalt model discussed before suggest that graded constraint satisfaction is an avenue worth pursuing.

Finally, it should be noted that the Miikkulainen and Dyer (1991) model fares similarly to the Story Gestalt model on each of the preceding criteria.
This result is not so surprising given the similarity in the architectures of the models. Both use a recurrent network with trainable weights to process propositions into schemas. The major difference in these two models is that the Story Gestalt model relies more heavily on learning to produce its representations. The Miikkulainen and Dyer model learns its own representation of concepts in its schemas through training, but the script-role structure of its schemas is stipulated. The Story Gestalt model learns these script roles for itself.

The Story Gestalt model also addresses a somewhat larger set of processing tasks, including pronoun resolution, revision, and generalization. Actually, Miikkulainen and Dyer's content + ID representation also produces good generalization to novel concepts when the content portion is the same as those trained, but the ID portion is different. The content portion is processed normally, and the ID portion is copied through the network. This solution, however, relies on stipulating this structure of concept representation and applying it to novel concepts. The Story Gestalt model further demonstrates how one environmental factor, the combinatorics of the corpus, leads to better or worse generalization.

CONCLUSIONS

The Story Gestalt model demonstrates the ability to perform a number of knowledge-intensive comprehension processes: representing the explicit text, resolving pronouns, drawing inferences, making revisions, and sharing information across contexts. It computes these different processes using a single, unified mechanism: graded constraint satisfaction. In this mechanism, a text is viewed as a set of constraints on its full or gestalt interpretation. The constraint-satisfaction process uses these constraints to compute a full interpretation of the text.

Graded constraint satisfaction has several properties which make it especially useful for computing these processes. First, it provides an easy way to combine multiple sources of information to resolve pronouns and draw inferences. Second, constraints can have strengths that can be used to represent the graded nature of much semantic information and that can be used to draw inferences to the degree they are supported by evidence in the text. Third, constraints can be easily added and subtracted as more text is processed, making revision and updating easier.

The graded nature of constraints also allows for some insight into generalization. In natural language, a key ingredient for good generalization is knowing the core meaning of each word. This information is represented in the model as identity constraints, and the semantic relations among concepts are associative constraints. Strong identity constraints produce the equivalent of slots in a script that can take on a large variety of values. This
ability is one of the key components of compositionality (Fodor & Pylyshyn, 1988). Importantly, though, these slots are not entirely free to take on any value: Instead, they are graded. Weaker or stronger associative constraints could make certain values harder or even impossible to represent and recall correctly. The relative strengths of these constraints determine the degree of schema regularization versus faithful recall, and the combinatorics of the training corpus is one factor that can affect their relative strengths.

The Story Gestalt model performs well but faces several challenges in representation and learning time. More importantly, the basic idea of graded constraint satisfaction as the underlying mechanism appears to be a sound and a productive way in which to understand knowledge-intensive processes in text comprehension.

REFERENCES

KNOWLEDGE INTENSIVE PROCESSES


APPENDIX: TEXT CORPUS

Three example scripts are provided here. The following is a description of how to read the scripts. At various points throughout a script there are choice points where the series of propositions diverges along different paths. Script paths are represented as a sequence of propositions at a common level of indentation (see the following). A choice point creates a set of sub-paths that are represented at the next-greater level of indentation and are enclosed in braces. Some roles have variables, such as person1 and person2, that are instantiated with specific characters when a text is built. In some cases, when a character is introduced in a text, there is a restriction on which characters can be instantiated. For example person = male restricts that person to be male.

There are two types of choice points: ands and ors. For an and, there is a probability that that path should be included in the text. If included, the path inside the brace is followed. When the closing brace is found, the path is finished, the level pops back to the level of indentation where the path started, and the text continues. If the and path is not chosen, the text-making process skips to the next proposition at the current tab level. The propositions in the and path, however, are remembered so that they can be added to the question list for training. For an or path, there is a set of paths and one is chosen to be followed instead of the others. An or path, like an and path, is represented at the next-greater tab level and is enclosed in braces.

Glossary

and [if variable = value] n—An and path. Optionally, there may be a restriction so that the path is chosen only if that variable has a specified value; n is the probability for choosing that path.

or j n—An or path; j is the number of alternate paths, and n is the probability for choosing this path.

Text: Beach .165 Restrictions: Camaro [

(agent = person1, predicate = decided-to-go, destination = beach)
(predicate = distance, location = beach, attribute = far)
or 2 .50 [
(agent = person1, predicate = got-in, destination = vehicle)
(agent = person1, predicate = drove, patient = vehicle, 
recipient = beach, manner = long)
and if person1 = male 1.00 [ 
(agent = person1, predicate = proceeded, patient = vehicle, 
destination = beach, manner = fast) 
and .50 [ 
(agent = policeman, predicate = gave, 
patient = ticket, recipient = person1) ]
]
(agent = person1, predicate = parked, patient = vehicle, 
location = beach, manner = free)
]
.50 [ 
(agent = person1, predicate = drove, patient = vehicle, 
destination = beach, manner = long)
]
and .00 [ 
(agent = person1, predicate = swan, location = beach) 
(agent = person1, predicate = won, theme = race, location = beach) ]
and if person1 = male .67 [ 
(agent = person 1, predicate = surfed, location = beach) 
(agent = person1, predicate = hung-ten) ]
and if person1 = female .33 [ 
(agent = person1, predicate = surfed, location = beach) 
and 0.0 [ 
(agent = person1, predicate = hung-ten) ]
]
and .33 [ 
(agent = person1, predicate = played, theme = volleyball, location = beach) ]
or 2
.80 [ 
(predicate = weather, location = beach, attribute = sunny) 
(agent = person1, predicate = returned, destination = home, 
manner = long) 
(predicate = mood, patient = person1, attribute = happy) ]
.20 [ 
(predicate = weather, location = beach, attribute = raining) 
(agent = person1, predicate = returned, destination = home, 
manner = long) 
(predicate = mood, patient = person1, attribute = sad) ]
]
Text: Airport .165 [  
(agent = person1, predicate = decided-to-go, destination = airport)  
(agent = person1, predicate = found, patient = change)  
(agent = person1, predicate = drove, patient = vehicle, destination = airport, manner = long)  
(agent = person1, predicate = ran, destination = gate, location = airport)  
(agent = person1, predicate = met, patient = person2, location = airport)  
(agent = person1-and-person2, predicate = returned, destination = home, manner = long)  
]

Text: Bar .165 Restrictions: Andrew Sarah [  
(agent = person1 = female, predicate = met, patient = person2 = male, location = bar)  
and if person1 = rich 1.00 [  
  (agent = person1, predicate = enjoyed, patient = expensive-wine, location = bar)  
]  
and if person1 = cheap 1.00 [  
  (agent = person1, predicate = enjoyed, patient = cheap-wine, location = bar, manner = not)  
]  
(agent = person2 = male, predicate = ordered, patient = drink, recipient = waiter, location = bar)  
and if person2 = rich 1.00 [  
  (predicate = quality, patient = drink, attribute = expensive)  
]  
and if person2 = cheap 1.00 [  
  (predicate = quality, patient = drink, attribute = cheap)  
]  
or 2  
  .50 [  
    (agent = person2, predicate = made, patient = pass, recipient = person1, location = bar, manner = politely)  
  or 2  
  .30 [  
    (agent = person1, predicate = gave, patient = slap, recipient = person2, location = bar)  
    (agent = person2, predicate = rubbed, patient = cheek, location = bar)  
  ]  
  .70 [  
    (agent = person1, predicate = gave, patient = kiss, recipient = person2, location = bar)  
    (agent = person2, predicate = rubbed, patient = lipstick, location = bar)  
  ]  
]
.50 [  
  (agent = person2, predicate = made, patient = pass,  
      recipient = person1, location = bar, manner = obnoxious)  
  or 2 
  .70 [  
    (agent = person1, predicate = gave, patient = slap,  
      recipient = person2, location = bar)  
    (agent = person2, predicate = rubbed, patient = cheek,  
      location = bar)  
  ]  
  .30 [  
    (agent = person1, predicate = gave, patient = kiss,  
      recipient = person2, location = bar)  
    (agent = person2, predicate = rubbed, patient = lipstick,  
      location = bar)  
  ]  
]