# Learning and Applying Contextual Constraints in Sentence Comprehension*

**Mark F. St. John** and James L. McClelland**
*Department of Psychology, Carnegie-Mellon University, Pittsburgh, PA 15213, USA*

ABSTRACT

*A parallel distributed processing model is described that learns to comprehend single clause sentences. Specifically, it assigns thematic roles to sentence constituents, disambiguates ambiguous words, instantiates vague words, and elaborates implied roles. The sentences are pre-segmented into constituent phrases. Each constituent is processed in turn to update an evolving representation of the event described by the sentence. The model uses the information derived from each constituent to revise its ongoing interpretation of the sentence and to anticipate additional constituents. The network learns to perform these tasks through practice on processing example sentence/event pairs. The learning procedure allows the model to take a statistical approach to solving the bootstrapping problem of learning the syntax and semantics of a language from the same data. The model performs very well on the corpus of sentences on which it was trained, and generalizes to sentences on which it was not trained, but learns slowly.*

## 1. Introduction

The goal of our research has been to develop a model that can learn to convert a simple sentence into a conceptual representation of the event that the sentence describes. Specifically, we have been concerned with the later stages of this process: the conversion of a sequence of sentence constituents, such as noun phrases, into a representation of the event. A number of problems make this process difficult. First, the words of a sentence may be ambiguous or vague. In the sentence, "The pitcher threw the ball," each content word is ambiguous. "Pitcher" could either refer to a ball-player or a container;

*The authors would like to thank Geoffrey Hinton, Brian MacWhinney, Andrew Hudson, and the members of the PDP Research Group at Carnegie-Mellon. This research was supported by NSF Grant BNS 86-09729, ONR Contracts N00014-86-G-0146, N00014-86-K-00167, and N00014-86-K-0349, and NIMH Career Development Award MH00385 to the second author.

**Present address: Department of Cognitive Science, University of California at San Diego, La Jolla, CA 92093, USA.

*Artificial Intelligence* 46 (1990) 217–257
0004-3702/90/$03.50 



"threw" could either refer to toss or host; and "ball" could refer to a sphere or a dance. How are the appropriate meanings selected so that a single, coherent interpretation of the sentence is produced? Vague words also present difficulties. In the sentences, "The container held the apples" and "The container held the cola," the word "container" refers to two different objects [1]. How does the context affect the interpretation of vague words?

A third problem is the complexity of assigning the correct thematic roles [9] to the objects referred to in a sentence. Consider:

(1) The teacher ate the spaghetti with the busdriver.
(2) The teacher ate the spaghetti with the red sauce.
(3) The busdriver hit the fireman.
(4) The busdriver was hit by the fireman.

In the first two examples, semantics play an important role. In the first sentence, it is the reader's knowledge that busdrivers are people that precludes the reader from deciding the busdriver is to be served as a condiment. Instead, it must be that both he and the teacher are eating the spaghetti. Semantic constraints work conversely in the second sentence. In the third sentence, semantics do not help determine who is the agent and who is the patient. Instead, word order determines the thematic role assignments. The busdriver is the agent because "the busdriver" is the pre-verbal constituent. Finally, in the fourth sentence, the influence of other morphological features can be seen. The passive verb tense and the "by" preposition, in conjunction with the word order, determine that the busdriver is the patient. Thematic role assignment, then, requires the joint consideration of a variety of aspects of the sentence.

A fourth problem for processing sentences is that a sentence may leave some thematic constituents implicit that are nevertheless present in the event. For example in sentences (1) and (2) above, the spaghetti was undoubtedly eaten with forks. Psychological evidence indicates that missing constituents, when strongly related to the action, are inferred and added to the description of the event. McKoon and Ratcliff [23] found, for example, that "hammer" was inferred after subjects read "Bobby pounded the boards together with nails."

Our model of the comprehension process centers on viewing the process as a form of constraint satisfaction. The surface features of a sentence, its particular words and their order and morphology, provide a rich set of constraints on the sentence's meaning. Each feature constrains the meaning in a number of respects. Conjunctions of features, such as word order and passive-voice morphology, provide additional constraints. Together, the constraints lead to a coherent interpretation of the sentence [17]. These constraints are not typically all-or-none. Instead, constraints tend to vary in strength: some are strong and others are relatively weak. An example adapted from Marcus [18] provides an illustration of the competition between constraints.

(1) Which dragon did the knight give the boy?

(2) Which boy did the knight give the dragon?
(3) Which boy did the knight give the sword?
(4) Which boy did the knight give to the sword?

Apparently, in the first two sentences, a weak syntactic constraint makes us prefer the first noun as the patient and the noun after the verb as the recipient. The subtle semantics in the second sentence, that knights don't give boys to dragons, does not override the syntactic constraint for most readers, though it may make the sentence seem ungrammatical to some. In sentence (3), a stronger semantic constraint overrides this syntactic constraint: swords, which are inanimate objects, cannot receive boys. Finally, in the fourth sentence, a stronger syntactic constraint overrides the semantics. It is clear from this example that constraints vary in strength and compete to produce an interpretation of a sentence. A good method for capturing this competition is to assign real-valued strengths to the constraints, and to allow them to compete or cooperate according to their strength.

Parallel distributed processing, or connectionist, models are particularly good for modeling this style of processing. They allow large amounts of information to be processed simultaneously and competitively, and they allow evidence to be weighted on a continuum [20, 22]. A number of researchers have pursued this idea and have built models to apply connectionism to sentence processing [5, 6, 35]. The development of this approach, however, has been retarded because it is difficult to determine exactly what constraints are imposed by each feature or set of features in a sentence. It is even more difficult to determine the appropriate strengths each of these constraints should have. Connectionist learning procedures, however, allow a model to learn the appropriate constraints and assign appropriate strengths to them.

To take advantage of this feature, learning was added to our list of goals. The model is given a sentence as input. From the sentence, the model must produce a representation of the event to which the sentence refers. The actual event that corresponds to the sentence is then used as feedback to train the model. But learning is not without its own problems. Several features of the learning task make learning difficult. One problem is that the environment is probabilistic. On different occasions, a sentence may refer to different events, that is, it may be referentially ambiguous. For example, a sentence like, "The pitcher threw the ball," may refer to either the tossing of a projectile or the hosting of a party. The robust, graded, and incremental character of connectionist learning algorithms leads us to hope that they will be able to cope with the variability in the environment in which they learn.

A second learning problem concerns the difficulty of learning the mapping between the parts of the sentence and the parts of the event [10, 26]. Learning the mapping is sometimes referred to as a boot-strapping problem since the meaning of the content words and significance of the syntax must be acquired



from the same set of data. To learn the syntax, it seems necessary to already know the word meanings. Conversely, to learn the word meanings it seems necessary to know how the syntax maps the words onto the event description. The connectionist learning procedure takes a statistical approach to this problem. Through exposure to large numbers of sentences and the events they describe, the mapping between features of the sentences and characteristics of the events will emerge as statistical regularities. For instance, in the long run, the learning procedure should discover the regularity that sentences beginning with "the boy" and containing a transitive verb in the active voice refer to events in which a young, male human participates as an agent. The discovery of the entire ensemble of such regularities provides a joint solution to the problems of learning the syntax and the meanings of words.

Some aspects of these goals have been addressed by our own earlier work [21, 30]. However, these previous models used a cumbersome *a priori* representation of sentences that proved unworkable (see [30] for discussion). Given the recent successes in using connectionist learning procedures to learn internal representations [12, 28], we decided to explore the feasibility of having a network learn its own representation of sentences.

A final characteristic of language comprehension we wanted to capture is sometimes called the principle of immediate update [3, 19, 34]. As each constituent of the sentence is encountered, the interpretation of the entire event is adjusted to reflect the constraints arising from the new constituent in conjunction with the constraints from constituents already encountered. Based on all of the available constraints, the model should try to anticipate upcoming constituents. It should also adjust its interpretation of preceding constituents to reflect each new bit of information. In this way, particular sentence interpretations may gain and lose support throughout the course of processing as each new bit of information is processed. This immediate update should be accomplished while avoiding the difficulty of performing backtracking.

In sum, the model addresses six goals:

- to disambiguate ambiguous words;
- to instantiate vague words;
- to assign thematic roles;
- to elaborate implied roles;
- to learn to perform these tasks;
- to immediately adjust its interpretation as each constituent is processed.

## 2. Description of the SG Model

### 2.1. Task

The model's task is to process a single clause sentence without embeddings into a representation of the event it describes. The sentence is presented to the

model as a temporal sequence of constituents. A constituent is either a simple noun phrase, a prepositional phrase, or a verb (including the auxiliary verb, if any). The information each of these sentence constituents yields is immediately used as evidence to update the model's internal representation of the entire event. This representation is called the sentence gestalt because all of the information from the sentence is represented together within a single, distributed representation; the model is called the Sentence Gestalt, or SG, model because it contains this representation. This general concept of sentence representation comes from Hinton's pioneering work [11]. From the sentence gestalt, the model can produce, as output, a representation of the event. This event representation consists of a set of pairs. Each pair consists of a thematic role and the concept that fills that role. Together, the pairs describe the event.

## 2.2. Architecture and processing

The model consists of two parts. One part, the sequential encoder, sequentially processes each constituent to produce the sentence gestalt. The second part is used to produce the output representation from the sentence gestalt.

### 2.2.1. *Producing the sentence gestalt*

To process the constituent phrases of a sentence, we adapted an architecture from Jordan [15] that uses the output of previous processing as input on the next iteration (see Fig. 1). Each constituent is processed in turn to update the sentence gestalt. To process a constituent, it is first represented as a pattern of activation over the *current constituent* units. Activation from these units projects to the first hidden unit layer and combines with the activation from the *sentence gestalt* created as the result of processing the previous constituent. The actual implementation of this arrangement is to copy the activation from the *sentence gestalt* to the *previous sentence gestalt* units, and allow activation to feed forward from there. Activation in the hidden layer then creates a new

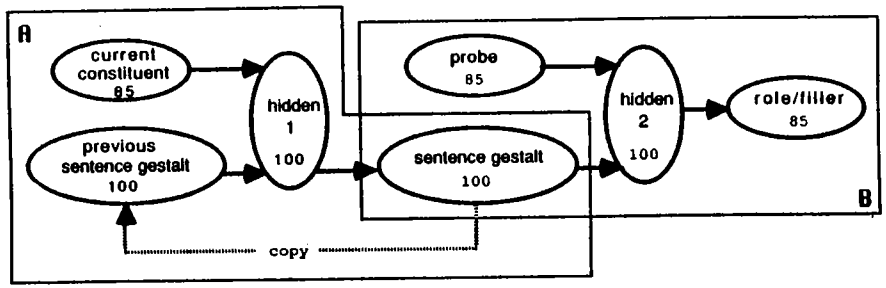


Fig. 1. The architecture of the network. The boxes highlight the functional parts: Area *A* processes the constituents into the sentence gestalt, and Area *B* processes the sentence gestalt into the output representation. The numbers indicate the number of units in each layer.



pattern of activation over the *sentence gestalt* units. The sentence gestalt, therefore, is not a superimposition of each constituent. Rather, each new pattern in the sentence gestalt is computed through two layers of weights and represents the model's new best guess interpretation of the meaning of the sentence.

### 2.2.2. *Producing the output*

As noted previously, several other models have used a type of sentence gestalt to represent a sentence. McClelland and Kawamoto [21] used units that represented the conjunction of semantic features of the verb with the semantic features of a concept. To encode a sentence, the patterns of activity produced for each verb/concept conjunction were activated in a single pool of units that contained every possible conjunction. St. John and McClelland [30] used a similar conjunctive representation to encode a number of sentences at once. These representations suffer from inefficiency and scale badly because so many units are required to represent all of the conjunctions. The current model's representation is far more efficient.

The model's efficiency comes from making the sentence gestalt a trainable, hidden unit layer. Making the sentence gestalt trainable allows the network to create the primitives it needs to represent the sentence efficiently. Instead of having to represent every possible conjunction, only those conjunctions that are useful will be learned and added to the representation. Further, these primitives do not have to be conjunctions between the verb and a concept. A hidden layer could learn to represent conjunctions between the concepts themselves or other combinations of information if they were useful for solving its task.

Since a layer of hidden units cannot be trained by explicitly specifying its activation values, we invented a way of "decoding" the sentence gestalt into an output layer. Backpropagation can then be used to train the hidden layer. The output layer represents the event as a set of thematic role and filler pairs. For example, the event described by "The pitcher threw the ball" would be represented as the set {agent/pitcher(ball-player), action/threw(toss), patient/ball(sphere)}. The words in parentheses indicate which concepts the ambiguous words correspond to.

The output layer can represent one role/filler pair at a time. To decode a particular role/filler pair, the sentence gestalt is probed with half of the pair, either the role or the filler. Activation from the probe and the *sentence gestalt* combine in the second hidden layer which in turn activates the entire role/filler pair in the output layer. The entire event can be decoded in this way by successively probing with each half of each pair.

When more than one concept can plausibly fill a role, we assume that the correct response is to activate each possible filler to a degree. The degree of activation of the units representing each filler corresponds to the filler's

conditional probability of occurring in the given context. The network should learn weights to produce these activations through training. To achieve this goal, we employed an error measure in the learning procedure, cross-entropy [13], that converges on this goal:

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

where $T_j$ is the target activation and $A_j$ is the output activation of unit $j$. As with many connectionist learning procedures, the goal is to minimize the error measure or cost-function [13]. The minimum of $C$ occurs at the point in weight space where the activation value of each output unit equals the conditional probability that the unit should be on in the current context. In the model, when the network is probed with a particular role, several of the output units represent the occurrence of a particular filler of that role. When $C$ is at its minimum, the units' activation values represent the conditional probability of the occurrence of that filler, in that role, given the current situation.[1] Note, however, that on any particular training trial, the target of training is one particular event. The minimum of $C$, then, is defined across the ensemble of training examples the model is shown.

Probing with the filler works similarly. The activation value of each role unit in the output layer represents the conditional probability of the probed filler playing that role in the current situation. In performing gradient descent in $C$, the network is searching for weights that allow it to match activations to these conditional probabilities.

### 2.3. Environment and training regime

Training consists of trials in which the network is presented with a sentence and the event it describes. The rationale is that the language learner experiences some event and then hears a sentence about it. The learner processes the sentence and compares the conceptual representation its comprehension mechanism produces to the conceptual representation it obtained from experiencing the event. Discrepancies are used as feedback for the comprehension mechanism. These sentence/event pairs were generated on-line for each training trial. Some pairs are more likely to be generated than others. Over training, these differences in the likelihood of generation translate into differences in training frequency.

The network is trained to generate the event from the sentence as input. To

[1] The situation, as defined in the learning procedure, is the combination of the previous sentence gestalt, the current constituent, and the current probe. It would be desirable to define the situation solely in terms of the sequence of sentence constituents. While our results suggest that the sentence gestalt learns to save all the relevant information from earlier constituents, we have no proof that it does.



promote immediate processing, a special training regime is used. After each constituent has been processed, the network is trained to predict the set of role/filler pairs of the entire event. From the first constituent of the sentence, then, the model is forced to try to predict the entire event. This training regime, therefore, assumes that the complete event is available to the learning procedure as soon as sentence processing begins, but it does not assume any special knowledge about which aspects of the event correspond to which sentence constituents. Of course, after processing only the first constituent, the model generally cannot correctly guess the entire event. By forcing it to try, this training procedure requires the model to discover the mapping between constituents and aspects of the event, as it forces the model to extract as much information as possible from each constituent. Consequently, as each new constituent is processed, the model's predictions of the event are refined to reflect the additional evidence it supplies.

### 2.4. An illustration of processing

An example of how a trained network processes a sentence will help illustrate how the model works. To process the sentence, "The teacher ate the soup," the constituents of the sentence are processed in turn. As each constituent is processed, the network performs a type of pattern completion. The model tries to predict the entire event by augmenting the information supplied by the constituents processed so far with additional information that correlates with the information supplied by the constituents.

With each additional constituent, the model's predictions improve. Early in the sentence, many possible events are consistent with what little is known about the sentence so far. The completion process activates each of these alternatives slightly, according to their support. As more constituents are processed, the additional evidence more strongly supports fewer possible events.

The pattern of activation over the sentence gestalt can be observed directly, and responses to probes can be examined, to see what it is representing after processing each constituent of the sentence (see Fig. 2). After processing the first constituent, "The teacher," of our example sentence, the network assumes the sentence is in the active voice and therefore assigns *teacher* to the agent role. The network also fills in the semantic features of teachers (person, adult, and female) according to its previous experience with teachers. When probed with the action role, the network weakly activates a number of possible actions which the teacher performs. The network similarly makes guesses about the other roles for which it is probed.

When the second constituent, "ate," is processed, the sentence gestalt is refined to represent the new information. In addition to representing both that *teacher* is the agent and that *ate* is the action, the network is able to make better guesses about the other roles. For example, it infers that the patient is

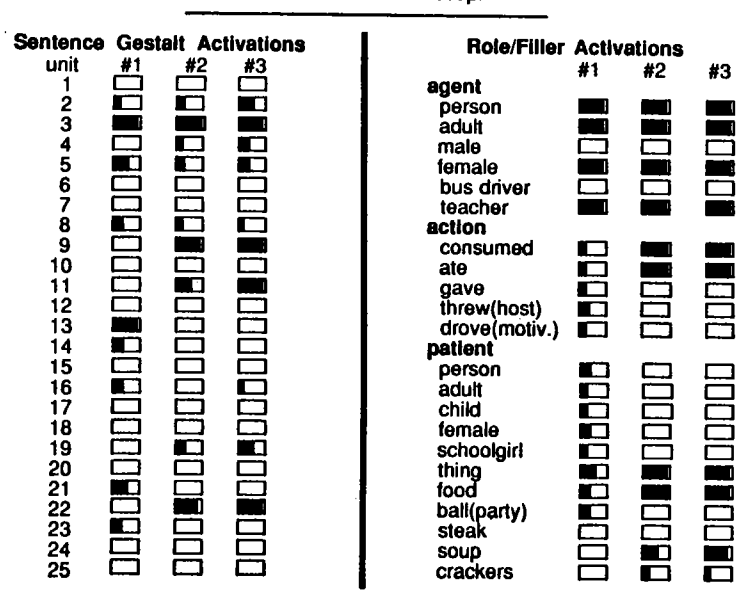


Fig. 2. The evolution of the sentence gestalt during processing. On the left, the activation of part of the sentence gestalt is shown after each sentence constituent has been processed. On the right, the activation of selected output units is shown when the evolving gestalt is probed with each role. The *#s* correspond to the number of constituents that have been presented to the network at that point. #1 means the network has seen "The teacher;" #2 means it has seen "The teacher ate;" etc. The activations (ranging between 0 and 1) are depicted as the darkened area of each box.

food. Since, in the network's experience, teachers typically eat soup, the network produces activation corresponding to the inference that the food is *soup*. After the third constituent is processed, the network has settled on an interpretation of the sentence. The thematic roles are represented with their appropriate fillers.

## 2.5. Specifics of the model

### 2.5.1. *Input representation*

Each sentence constituent can be thought of as a surface role/filler pair. It consists of one unit indicating the surface role of the constituent and one unit representing each word in the constituent. One unit stands for each of 13 verbs, 31 nouns, 4 prepositions, 3 adverbs, and 7 ambiguous words. Two of the ambiguous words have two verb meanings, three have two noun meanings, and two have a verb and a noun meaning. Six of the words are vague terms (e.g. someone, something, and food). For prepositional phrases, the preposition and



the noun are each represented by a unit in the input. For the verb constituent, the presence of the auxiliary verb "was" is likewise encoded by a separate unit. Articles are not represented, and nouns are assumed to be singular and definite throughout.

The surface role, or location, of each constituent is coded by four units that represent location respective to the verb: pre-verbal, verbal, first-post-verbal, and n-post-verbal. The first-post-verbal unit is active for the constituent immediately following the verb, and the n-post-verbal unit is active for any constituent occurring after the first-post-verbal constituent. A number of constituents, therefore, may share the n-post-verbal position. The sentence, "The ball was hit by someone with the bat in the park," would be encoded as the following ordered set in which the words in parentheses represent units in the input that are on for each constituent {(pre-verbal, ball), (verbal, was, hit), (first-post-verbal, by, someone), (n-post-verbal, with, bat), (n-post-verbal, in, park)}. Without the surface roles, the network must learn to use the temporal order of the constituents to produce syntactic constraints. Additional simulation has shown that the network can learn the corpus without the surface roles. Interestingly, removing the surface roles did not slow down learning.

### 2.5.2. *Output representation*

The output has one unit for each of 9 possible thematic roles (e.g., agent, action, patient, instrument) and one unit for each of 45 concepts, including 28 noun concepts, 14 actions, and 3 adverbs. Additionally, there is a unit for the passive voice. Finally, there are 13 "feature" units, such as male, female, and adult. These units are included in the output to allow the demonstration of more subtle effects of constraints on interpretation (see Appendix A for the complete set of roles and concepts). This representation is not meant to be comprehensive. Instead, it is meant to provide a convenient way to train and demonstrate the processing abilities of the network. Any one role/filler pattern, then, consists of two parts. For the role, one of the 9 role units should be active, and for the filler, a unit representing the concept, action, or adverb should be active. If relevant, some of the feature units or the passive voice unit should be active.[2]

### 2.5.3. *Training environment*

While the sentences often include ambiguous or vague words, the events are always specific and complete: each event consists of a specific action and each

[2] A second output layer was included in the simulations. This layer reproduced the sentence constituent that fits with the role/filler pair being probed. Consequently, the model was required to retain the specific words in the sentence as well as their meaning. Since this aspect of the processing does not fit into the context of the current discussion, these units are not discussed further. Additional simulation has shown that this extra demand on the network does not qualitatively affect its performance.

thematic role related to this action is filled by some specific concept. Accordingly, each event occurs in a particular location, and actions requiring an instrument always have a specific instrument.

Sentence/event pairs are created on-line during training from scaffoldings called sentence-frames. The sentence-frames specify which thematic roles and fillers can be used with that action. Each of the 14 actions has a separate sentence-frame. Four additional frames were made to cover passive versions of sentences involving the actions *kissed*, *shot*, *hit*, and *gave*.

To create a sentence/event pair, a sentence-frame is picked at random and then each thematic role is processed in turn. (Appendix B contains a sample sentence-frame.) For example, let's assume that the *Ate* sentence-frame is chosen. Agent is the first role processed. First, a concept to fill the role is selected from the set of concepts that can play the agent role in the *Ate* sentence-frame. This role/filler pair is added to the event description. Since some roles, such as instrument and location, may not be mentioned in the sentence, it is randomly determined, according to a preset probability, whether a role will be included in the sentence. If the role is to be included, a word is chosen to represent the filler in the sentence. Otherwise, the role is left out of the sentence, but it is still included in the event description. Since the agent role must be included in sentences about eating, it is placed in the sentence, and a word is chosen. Assuming *busdriver* is chosen as the filler concept, a word to describe *busdriver* is selected. For example, the word "someone" might be chosen.

Next, the action role is processed. Since the *Ate* sentence-frame is being used, the action must be *ate*. A word to describe *ate* is then chosen: "consumed," for example. Then the patient is chosen. The probabilities of choosing particular patients depend upon what has been selected for the agent and action. Given the selection of *busdriver* as the agent, *steak* is a much more likely patient than *soup*. Let's assume that *steak* is selected, and that the word "steak" is chosen to represent it. In general, by changing the probabilities of selecting specific fillers according to which other fillers have been selected so far, statistical regularities among the fillers will develop across the corpus.

In the same way, the remaining role/filler pairs for the sentence-frame are generated. Assuming only the first three roles are chosen to be included in this sentence, the input sentence will be, "Someone consumed the steak." The event will be the entire set of role/filler pairs {agent/busdriver, action/ate, patient/steak, instrument/knife, location/living-room, etc.}.

In this way, 120 different events can be generated from the set of frames with some being more likely to appear than others. The most frequent event occurs, on average, 5.5 times per 100 trials, but the least frequent event occurs only 9 times per 10,000. The number of words that can be chosen to describe an event and the option to include or eliminate optional constituents from the sentence brings the number of sentence/event pairs to 22,645.



In training the model on this corpus, sentence-frames were picked at random, and sentence/event pairs were generated according to the random procedure described above. No specific sentences or events were set aside to not be trained.

The sentences are limited in complexity because of the limitations of the event representation. Only one filler can be assigned to a role in a particular sentence. Also, all the roles are assumed to belong to the sentence as a whole. Therefore, no embedded clauses or phrases attached to single constituents are possible.

#### 2.5.4. *Training procedure details*

Each training trial consists of first generating a sentence/event pair, and then presenting the sentence to the model for training. The constituents are presented to the model sequentially, one at a time. After the model processes a constituent, the model is probed with each half of each role/filler pair for the entire event. The error produced on each probe is collected and propagated backward through the network (cf. [28]). The weight changes from each sentence trial are added together and used to update the weights after every 60 trials. The learning rate, $\epsilon$, was set to 0.0005, and momentum was set to 0.9. No attempt was made to optimize these values, so it is likely that learning time could be improved by tuning these parameters.

## 3. Results

### 3.1. Overall performance

First, we will assess the model's ability to comprehend sentences generally. Then we will examine the model's ability to fulfill our specific processing goals, and we will examine the development of the model's performance across training trials. Finally, we will discuss the model's ability to generalize.

Once the model was able to process correctly both active and passive sentences, the simulation was stopped and evaluated. Correct processing was defined as activating the correct units more strongly than the incorrect units. After 330,000 random sentence trials, the model began correctly processing the passive sentences in the corpus.

A set of 100 test sentence/event pairs were generated randomly from the corpus. These sentence/event pairs were generated in the same way the training sentences were generated except that they were generated without regard to their frequency during training, so seldom practiced pairs were as likely to appear in the test set as frequently practiced pairs. Of these pairs, 45 were set aside for separate analysis because they were ambiguous: at least two different interpretations could be derived from each (e.g. Someone ate something). Of the remaining sentence/event pairs, every sentence contained at

least one vague or ambiguous word, yet each had only one interpretation. These unambiguous sentence/event pairs were tested by first allowing the model to process all of the constituents of the sentence. Then the model was probed with each half of each constituent that was mentioned in the sentence. The output produced in response to each probe was compared to the target output. Figure 3 presents a histogram of the results.

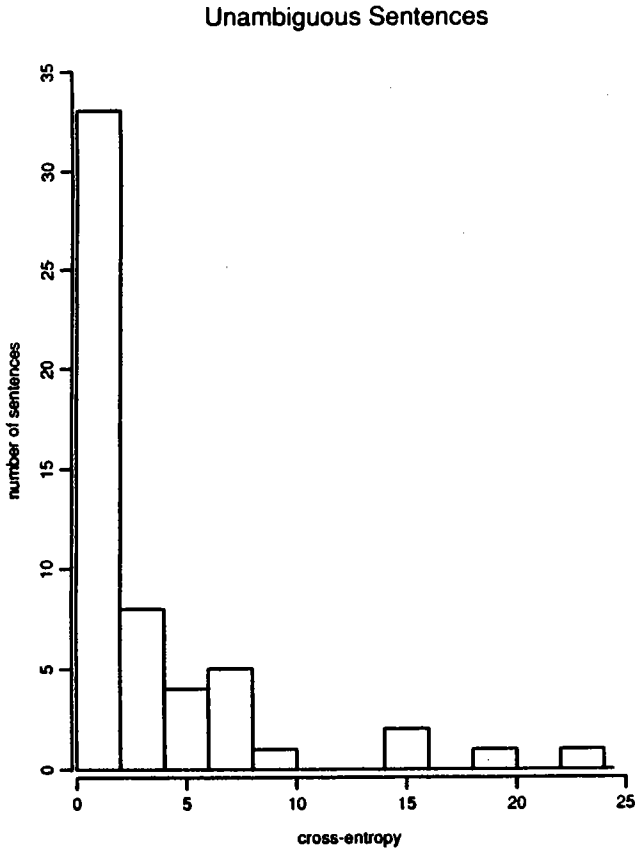


Fig. 3. Histogram of the cross-entropy error for random sentences after 330,000 sentence trials. The sentences were drawn randomly from the corpus without regard to their frequency. A cross-entropy measure of between 0 and 10 results from sentences that are processed almost perfectly. Only small errors occur when an output unit should be completely activate (with a value of 1), but only obtains an activation of 0.7 or 0.8, or when a unit should have an activation of 0, but has an activation of 0.1 or 0.2. Cross-entropy errors of between 15 and 20 occur when one of the role/filler pairs is incorrect. For example, if *teacher* were supposed to be the agent, but the network activates *busdriver*, an error of about 15 would result.



For these unambiguous sentences, the cross-entropy, summed over constituents, averaged 3.9 per sentence. Another measure of performance is the number of times an output unit that should be on is less active than an output unit that should be off. The idea behind this measure is that as long as the correct unit within any set, such as people or gender, is the most active, it can win a competition with the other units in that set. Checking that all of the correct units are more active than any of the incorrect units is a quick, and conservative, way of calculating this measure. An incorrect unit was more active in 14 out of the 1710 possible cases, or on 0.8% of the opportunities.

The 14 errors were distributed over 8 of the 55 sentences. In 5 of the 8 sentences, the error involved the incorrect instantiation of the specific concept, or a feature of that concept, referred to by a vague word. Two other errors involved the incorrect activation of the concept representing a nonvague word. In each case, the incorrect concept was similar to the correct concept. Therefore, errors were not random; they involved the misactivation of a similar concept or the misactivation of a feature of a similar concept. The errors in the remaining sentence involved the incorrect assignment of thematic roles in a passive, reversible sentence: "Someone hit the pitcher" (see the section on learning for a discussion of this problem).

Additional practice, of course, improved the model's performance. Improvement is slow, however, because the sentences processed incorrectly are relatively rare. After a total of 630,000 trials, the number of sentences having a cross-entropy higher than 15 dropped from 3 to 1. The number of errors dropped from 14 to 11.

### 3.2. Performance on specific tasks

Our specific interest was to develop a processor that could correctly perform several important language comprehension tasks. Five typical sentences were drawn from the corpus to test each processing task. The categories and one example sentence for each are presented in Table 1. The parentheses denote the implicit, to be inferred, role.

Table 1
Task categories

| Category | Example |
|---|---|
| Role assignment | |
| Active semantic | The schoolgirl stirred the kool-aid with a spoon. |
| Active syntactic | The busdriver gave the rose to the teacher. |
| Passive semantic | The ball was hit by the pitcher. |
| Passive syntactic | The busdriver was given the rose by the teacher. |
| Word ambiguity | The pitcher hit the bat with the bat. |
| Concept instantiation | The teacher kissed someone. |
| Role elaboration | The teacher ate the soup (with a spoon). |

The first category involves role assignment. The category was divided into four sub-categories based on the type of information available to help assign the correct thematic roles to constituents. Sentences in the active semantic group contain semantic information that can help assign roles. In the example from Table 1, of the concepts referred to in the sentence, only the schoolgirl can play the role of an agent of stirring. The network can therefore use that semantic information to assign schoolgirl to the agent role. Similarly, kool-aid is something that can be stirred, but cannot stir or be used to stir something else. After each sentence was processed, the sentence gestalt was probed with the filler half of each role/filler pair. The network then had to complete the pair by filling in the correct thematic role. For each pair, in each sentence, the unit representing the correct role was the most active. Sentences in the passive semantic category are processed equally well. Of course the semantic knowledge necessary to perform this task is never provided in the input or programmed into the network. Instead, it must be developed internally in the sentence gestalt as the network learns to process sentences.

Syntactic information does not have to be used in these cases; the semantic constraints suffice. In fact, if the surface location of the constituents is removed from the input, the roles are still assigned correctly. Further, if the constituents are presented in different orders, the activation values in the output are affected only slightly. Sentence processing that can rely on semantic information, therefore, essentially does, though confusing the syntax appears to have a slight corrupting effect.

The relative strengths of syntactic and semantic constraints are determined by their reliability in the training corpus. The more reliable a constraint, the more potent its influence in processing. This effect of the corpus on later processing is also found in natural languages. Word order in English is very reliable and is a very strong constraint on the meaning of a sentence. In Italian, however, word order is less reliable and is a much weaker constraint which can be over-ridden by semantic constraints. Consequently, the sentence "The pencil kicked the cow" in English is taken to mean that the pencil did the kicking because of the word order, while in Italian it is taken to mean that the cow did the kicking because of the semantics of the situation [17].

To process sentences in the active and passive syntactic categories, however, the network cannot rely entirely on semantic constraints to assign thematic roles. Sentences in these categories were created by including in the corpus pairs of reversible events, such as the busdriver giving a rose to the teacher, and the teacher giving a rose to the busdriver. Both of these events were trained with equal frequency. Without a difference in frequency, there is no semantic regularity to help predict which of the two events a sentence refers to. The model must rely on syntactic information, such as word order, to assign the thematic roles. Passive sentences further complicate processing by making word order, by itself, unpredictive. The past participle and the "by" preposition provide cues designating the passive, but in themselves do not cue which person plays which role either. The word order information must be used in conjunction with the passive cues to determine the correct role assignments.



When sentences in the syntactic categories were tested, for each role/filler pair in each test sentence, the correct role was the most active. Figure 4 provides an example of role assignment in the semantic and syntactic categories.

The remaining three categories involve the use of context to help specify the concepts referred to in a sentence. Sentences in the word ambiguity category contain one or more ambiguous words. After processing a sentence, the network was probed with the role half of each role/filler pair. The output patterns for the fillers were then examined. Figure 5 provides an example sentence with ambiguous words. For all pairs in each test sentence, the correct filler was the most active.

Disambiguation requires the competition and cooperation of constraints from both the word and its context. While the word itself cues two different interpretations, the context fits only one. In "The pitcher hit the bat with the bat," "pitcher" cues both *container* and *ball-player*. The context cues both *ball-player* and *busdriver* because the model has seen sentences involving both people hitting bats. All the constraints supporting *ball-player* combine, and

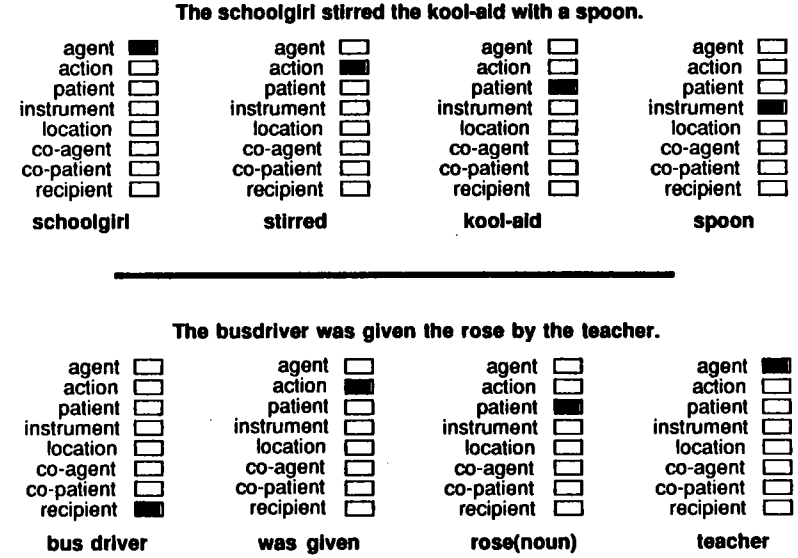


Fig. 4. Role assignment. After a sentence is processed, the network is probed with the filler half of each role/filler pair. The activation over a subset of the thematic role units is displayed. The first sentence contains semantic information useful for role assignment, while the second sentence contains only syntactic information useful for role assignment.

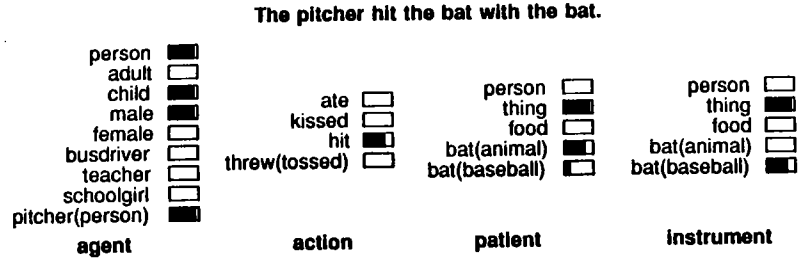


Fig. 5. Word disambiguation. The sentence "The pitcher hit the bat with the bat" is processed by the network. The network is then probed with each thematic role in the event. The activation over a subset of the fillers is displayed. The network correctly disambiguates each word.

together they win the competition for the interpretation of the sentence. As can be seen from the present example, even when several words of a sentence are ambiguous, the event which they support in common dominates the disparate events that they support individually. The processing of both instances of "bat" work similarly: the word and the context mutually support the correct interpretation. Consequently, the final interpretation of each word fits together into a globally consistent event.

Concept instantiation works similarly. Though the word cues a number of more specific concepts, only one fits the context. Again, the constraints from the word and from the context combine to produce a unique, specific interpretation of the term. As with the disambiguation task, each test sentence was processed, and then the network was probed with the role half of each role/filler pair. The output filler patterns were examined to see if the correct concept and semantic features were instantiated (see Fig. 6). In each case, the correct concept and features were the most active.

Depending upon the sentence, however, the context may only partially constrain the interpretation. Such is the case in "The teacher kissed someone." "Someone" could refer to any of the four people found in the corpus. Since, in the network's experience, females only kiss males, the context constrains the interpretation of "someone" to be either the busdriver or the pitcher, but no further. Consequently, the model can activate the *male* and *person* features of the patient while leaving the units representing *busdriver* and *pitcher* only partially active. The features *adult* and *child* are also partially and equally active because the busdriver is an adult while the pitcher is a child (see Fig. 6). While *pitcher* is slightly more active in this example, neither is activated above 0.5 (see the section on ambiguous sentences for an explanation of the difference in activations). In general, the model is capable of inferring as much information as the evidence permits: the more evidence, the more specific the inference.

Word disambiguation can be seen as one type of this general inference



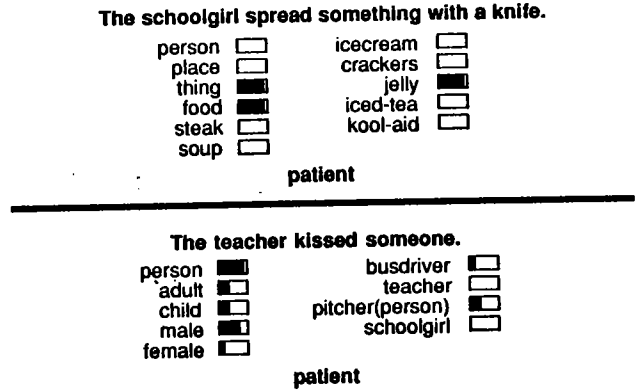


Fig. 6. Concept instantiation. The network has learned that *jelly* is always the patient of *spread*. When the network processes "The schoolgirl spread something with a knife," it instantiates "something" as *jelly*. For the sentence "The teacher kissed someone," the network partially instantiates "someone" as a *male* and *person*, and activates both *pitcher* and *busdriver* partially.

process. The only difference is that for ambiguous words, both the general concept and the specific features differ between the alternatives, while for vague words, the general concept is the same and only some of the specific features differ.

Finally, sentences in the role elaboration category test the model's ability to infer thematic roles not mentioned in the input sentence. For example, in "The teacher ate the soup," no instrument is mentioned, yet a spoon can be inferred. For each test sentence, after the sentence was processed, the network was probed with the role half of the to-be-inferred role/filler pair. The correct filler was the most active in each case. Figure 7 provides an example. For role

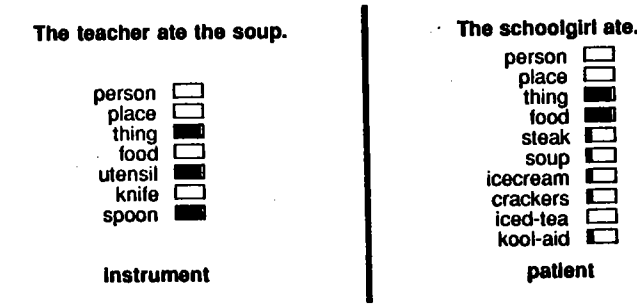


Fig. 7. Role elaboration. After processing the sentence "The teacher ate the soup," the network is probed with the instrument role. The filler activations are displayed. The network correctly infers *spoon*. For "The schoolgirl ate," the model must infer a patient. Because the schoolgirl is likely to eat a variety of foods, no particular food is well activated.

elaboration, the context alone provides the constraints for making the inference. Extra roles that are very likely will be inferred strongly. When the roles are less likely, or could be filled by more than one concept, they are only weakly inferred.

As it stands, there is nothing to keep the network from generalizing to infer extra roles for every sentence, even for events in which these roles make no sense. For instance, in "The busdriver drank the iced-tea," no instrument should be inferred, yet the network infers knife because of its association with the busdriver. It appears that since the busdriver uses a knife in many events about eating, the network generalizes to infer the knife as an instrument for his drinking. However, in events further removed from eating, instruments are not inferred. For example, in "The busdriver rose," no instrument is activated. It appears, then, that generalization of roles is affected by the degree of similarity between events. When events are similar, elaborative roles may be generalized. When events are distinct, roles do not generalize, and the model has no reason to activate any particular filler for a role.

### 3.3. Immediate update

As each constituent is processed, the information it conveys modifies the sentence gestalt and strengthens the inferences it supports. But the beginning of a sentence may not always accurately predict its eventual full meaning. For example, in "The adult ate the steak with daintiness," the identity of the adult is initially unknown. After "The adult ate" has been processed, *busdriver* and *teacher* are equally active. After processing "The adult ate the steak," the model guesses that the agent is the *busdriver* since steak is typically eaten by busdrivers. At this point, the model has sufficient information to instantiate "the adult" to be the *busdriver*. Along with this inference, *gusto* is inferred as the manner of eating, since busdrivers eat with gusto. Here the model demonstrates its ability to infer additional thematic roles.

The model has, at this point, been led down the garden path toward an ultimately incorrect interpretation of the sentence. The next constituent processed, "with daintiness," only fits with the teacher and the schoolgirl. Since the sentence specifies an adult, the agent must be the *teacher*. The model must revise its representation of the event to fit with the new information by de-activating *busdriver* and activating *teacher* (see Fig. 8).

In general, as each constituent is processed, the information it explicitly conveys is added to the representation of the sentence along with implicit information implied by the constituent in the current situation. When the evidence is ambiguous and supports may conflicting inferences (such as after "The adult ate" has been processed) all the inferences are weakly activated in the sentence gestalt. When new evidence suggests a different interpretation, the sentence gestalt is revised.



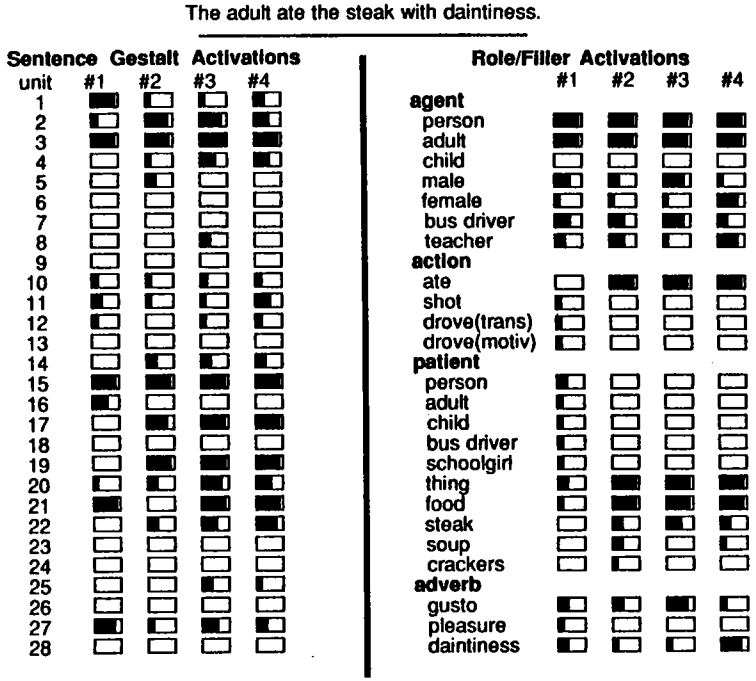


Fig. 8. The sequential processing of a garden-path sentence. After "the steak" has been processed, the network instantiates "the adult" with the concept *busdriver*. When "with daintiness" is processed, the network must reinterpret "the adult" to mean *teacher*.

### 3.4. Ambiguous sentences

The ambiguous sentences in the test set were tested separately. As noted above, an ambiguous sentence has more than one consistent interpretation. For example, the adult in the sentence, "The adult drank the iced-tea in the living-room," can be instantiated with either *busdriver* or *teacher* as the agent, but the sentence offers no clues that *teacher* is the correct agent in this particular sentence/event pair in the test set. In these ambiguous cases, the model should compromise and activate *busdriver* and *teacher* partially and equally, causing two small errors. What the network typically did, however, was to activate one concept slightly more than the other.

One reason for these differences in activations is the recent training history of the network. The sentence/event pairs trained more recently have a greater impact on the weights and, therefore, on subsequent processing. Because selection of training examples occurs randomly, several sentences involving a particular agent may occur before a sentence/event involving a different agent

is trained. Such training biases can lead to a bias in the activation of alternatives in ambiguous sentences. We tested this explanation by training the network on sentence/event pairs that consisted of an ambiguous sentence and the subordinate, weakly activated, event. From one to three training trials were required to balance the activation of the subordinate event with that of the previously dominant event.

The sensitivity of the network to recent training on ambiguous sentences is due to the dynamics of the activation function. Because the activation function is sigmoidal, it is sensitive to changes in the value of its input when the value is in the middle of its range. Since each meaning of an ambiguous sentence should be activated partially, its inputs to the activation function must lie in this middle range. Consequently, minor changes to the weights that determine the input value will have a major impact on the activation value.

Conversely, in the extremes of the range of the activation function, changes in the input value will have little discernible effect on the activation value. Since the one meaning of an unambiguous sentence should be activated fully, its inputs to the activation function lie in the extremes of the function's range. Minor changes in the weights, therefore, will have only minor changes on the activation value, making the processing of unambiguous sentences robust to recent training.

### 3.5. Learning

As the network learns to comprehend sentences correctly, a number of developmental phenomena can be observed. In fact, the only real failures in performance stem from a developmental effect. Problems in processing only arise in processing infrequent and irregular sentences. For example, sentences about the busdriver eating soup are rare. The network is seven times more likely to see a sentence about the busdriver eating steak than eating soup. This frequency difference creates a strong regularity between "The busdriver ate" and the concept *steak*. In a sentence about the busdriver eating soup, the word "soup" constrains the patient to be *soup*, while "The busdriver ate" partially constrains the patient to be *steak*. The constraints compete for an interpretation of the sentence. When the regularities are particularly strong, the contextual constraints can win the competition and cause the bottom-up activation from the word itself to be overridden.

Though this effect seems like a serious flaw, it is a flaw that the model shares with people. In an illuminating experiment, Erickson and Mattson [8] asked subjects questions like, "How many animals of each kind did Moses take on the Ark?" Subjects typically answered, "Two," despite their knowledge, when later asked, that Moses had nothing to do with the Ark. Constraints from the context overwhelmed the constraint from the word "Moses."

Erickson and Mattson also describe a second order effect in that subjects will



balk when asked, "How many animals of each kind did Nixon take on the Ark?" Apparently, the degree of semantic overlap between the correct concept and the foil affects how easily subjects will be misled. The model demonstrates the second order effect as well. Given "The busdriver ate the ball," the model fails to activate any patient. The model cannot explicitly balk, but its failure to represent the sentence accurately or misinterpret it is similar to balking. This example suggests that the SG model holds promise for demonstrating interesting human-like errors in comprehension.

In the model, this frequency or regularity effect diminishes with training: the reliability of a constraint, its probability of correctly predicting the output, rather than its overall frequency, becomes increasingly important. The word "soup" perfectly predicts the concept *soup*: whenever "soup" appears in a sentence, the event contains the concept *soup*. On the other hand, the busdriver eats a variety of foods: "the busdriver ate" is only 70% reliable as a predictor of *steak*. With increased training, even low frequency constraints are practiced. If they are reliable, they gain strength and eventually outweigh more frequent but less reliable constraints. Similar developmental trends occur as children learn language [17]. Progress is slow, but after a total of 630,000 trials even these very infrequent and irregular sentences are processed correctly.

The early effect of frequency works for syntactic constraints as well as semantic constraints. As shown in Fig. 9, the model masters sentences in the active voice sooner than it masters sentences in the passive voice. This difference is due to the greater frequency of sentences in the active voice in the corpus. While 14 sentence frames use the active voice, only 4 frames use the passive voice. After 330,000 trials, though, both voices are handled correctly.

The syntactic constraints develop more slowly than the regular semantic constraints. Yet while every sentence contains word order constraints, only an occasional sentence will contain a particular semantic constraint. Based on the

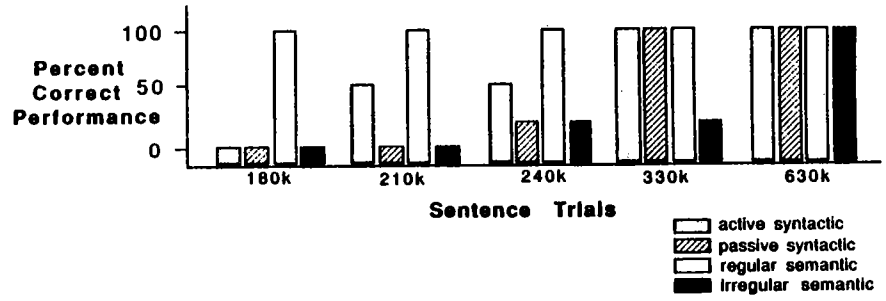


Fig. 9. Development of performance. active syntactic—The busdriver kissed the teacher; passive syntactic—The teacher was kissed by the busdriver; regular semantic—The busdriver ate the steak; irregular semantic—The busdriver ate the soup. Correct performance means the correct concepts are more active than incorrect concepts.

frequency of practice with particular constraints, then, the word order constraints should be learned much earlier than the semantic constraints. Two caveats to the frequency rule help explain this result. First, the syntactic constraints involve the conjunction of word order with the presence or absence of the passive markers, and such conjunctions are difficult to learn. Second, learning tends to generalize across semantically similar words, so training on one word can facilitate the learning of similar words.

The large number of training trials required to achieve good performance led us to look for ways to improve the speed of learning. One experiment consisted of removing the first hidden layer from the architecture. The input and the previous sentence gestalt then fed directly into the sentence gestalt layer. It was hoped that by making the network one layer more shallow, and by reducing the number of weights, error correction would proceed more quickly. Training of the modified network was stopped when it became apparent that the model did not learn the semantic constraints more quickly, and it had not learned the syntactic constraints at all. It is possible that the network could still learn the syntactic constraints given more training. The point of the modification, however, was to speed learning, and this goal was not fulfilled.

The reason for the utility of the first hidden layer in computing the syntactic constraints lies in the conjunctive nature of the constraints. The hidden layer is useful in computing the conjunction of word order and active/passive voice markers. Without the hidden layer, the sentence gestalt would have to compute the conjunction as well as represent the meaning of the sentence. Such a representation is apparently difficult for the network to find.

### 3.6. Representations

While the input to the network is a local encoding where each word is represented by a different unit, the network can create internal representations that are distributed and that explicitly encode helpful semantic information. The weights running from the input layer to the first hidden layer can be seen as "constraint vectors" which determine how each word influences the evolution of the sentence gestalt. These constraint vectors are the model's bottom-up representation of each word. Words that impose similar constraints should develop similar constraint vectors. A cluster analysis of the weight vectors reveals their similarity. Separate cluster analyses were performed for unambiguous verbs and nouns (see Fig. 10).

The verbs cluster into a number of hierarchical groups. One cluster contains the consumption verbs. Another contains stirred and spread. These two clusters then combine into a cluster of verbs involving people and food. Kissed, hit, and shot formed another cluster. For each of these verbs there were passive-voice sentences in the corpus, and each could take an animate object. Gave, the only dative verb, stands apart from the other verbs. This clustering



reflects the similarity of the case frames of the members of the different clusters.

The constraint vectors of nouns further reflect similarities in the constraints they impose on the evolving sentence gestalt. This similarity is reflected in two ways. As with the verbs, semantically similar words cluster: all of the people cluster, and dog and spot are very similar. Words that occur together in the same context also have similar constraint vectors. For example, ice cream clusters with park, and jelly clusters with knife. In the corpus, ice cream is always eaten in the park, and jelly is always spread with a knife. Their similar constraint vectors follow from the similar constraints they impose on the events described by the sentences in which they appear.

### 3.7. Generalization

An important remaining question is whether the model is actually learning useful constraints that it can apply to novel sentences or whether it is simply memorizing sentence/event pairs. Since sentences in the first simulation were

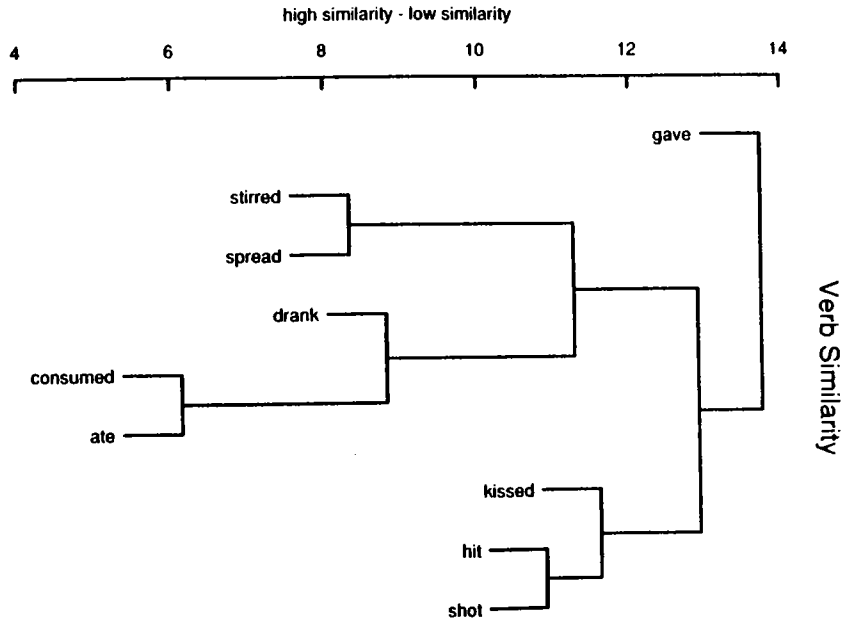


Fig. 10. Cluster analyses. The analysis computes the similarity between the weight vectors leading from each input unit to the first hidden layer. The more similar two vectors or clusters of vectors (in euclidean distance), the sooner they are combined into a new cluster. Physical distance in the figure is irrelevant; only the clustering is important. For instance, "stirred" is not notably more similar to "drank" than it is to "ate."

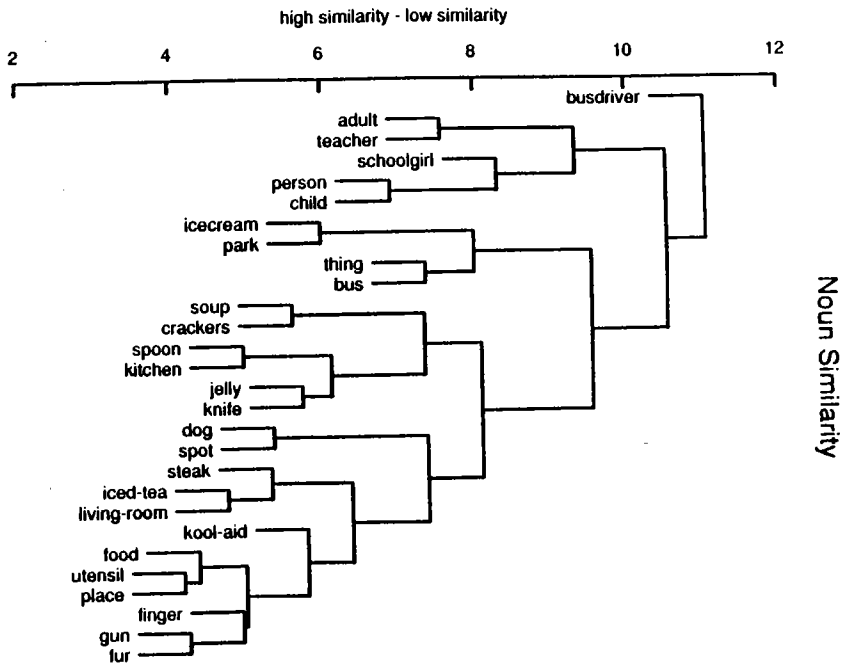


Fig. 10 (contd.)

generated randomly and none were set aside to not be trained, the first simulation cannot be used to evaluate generalization. Two new corpora were developed to test the model's generalization abilities. One of the new corpora was designed to test the model's ability to generalize syntactic regularities and the other was designed to test its ability to generalize semantic regularities. The model was trained and evaluated on each corpus separately.

### 3.7.1. *Syntax*

For syntax, we tested the model's ability to learn and use the syntax of active and passive sentences. Could the model learn to use the active/passive voice markers and the temporal order of the constituents, word order, productively on novel sentences? This is a test of compositionality. The model must learn to compose the familiar constituents of sentences in new combinations. To perform this generalization task, the model must learn several types of information. First, it must learn the concept referred to by each word in a sentence: "John" refers to *John* and "saw" refers to *saw*. Second, the model must learn that the order of the constituents, which constituent comes before the verb and which after, is important to assigning the agent and patient. Third, the model must learn the relevance of the passive markers: the past



participle of the verb and the prepositional phrase beginning with "by." Fourth, the model must learn to integrate the order information and the passive marker information to correctly assign the thematic roles.

The model was trained on a corpus of sentences composed of ten people and ten reversible actions, such as "John saw Mary." These sentences could appear in the active or passive voice. The basic corpus consisted of all 2000 sentences (10 people by 10 actions by 10 people by 2 voices). Before training began, though, 250 of these sentences (12.5%) were set aside for later testing: they were not trained. The model was then trained on the remaining sentences. Once the training corpus was mastered (after 100,000 trials), the 250 test sentences were presented to the model. The model processed 97% of these sentences correctly. In only 11 sentences did the model incorrectly assign a thematic role. In these sentences, when probed with one of the fillers, the model activated an incorrect role more strongly than the correct role.

### 3.7.2. *Semantics*

Semantic regularities may also provide the basis for generalization. We tested the model's ability to learn some semantic regularities and apply them in novel contexts. There are two basic generalization effects we would like to see in the model's behavior. First, as with syntax, the trained model should exhibit compositionality: the model should be able to represent successfully sentences it has never seen before. Secondly, the model should make predictions: it should be able to use information presented early in a sentence to help it process subsequent information.

To test these generalization effects, we again created a simple corpus for the model to learn. This corpus consisted of 8 adults and 8 children, 5 actions, and 8 objects for each action. The corpus was arranged to make age a semantic regularity. For each action, 3 objects were presented with adults, 3 were presented with children, and 2 were presented with both adults and children. This arrangement created a corpus of 400 sentences (8 adults by 5 actions by 5 objects plus 8 children by 5 actions by 5 objects). For example,

| | |
|---|---|
| George watched the news. | Bobby watched He-Man. |
| George watched Johnny Carson. | Bobby watched Smurfs. |
| George watched David Letterman. | Bobby watched Mighty Mouse. |
| George watched Star Trek. | Bobby watched Star Trek. |
| George watched the Road Runner. | Bobby watched the Road Runner. |

The regularity is that the age of the agent, in conjunction with the verb, predicts a set of objects. Age, though, is not explicitly encoded in the input or the output representations. Instead, it is a "hidden" feature (Unlike people in

the first simulation, people in this corpus are represented by a single unit, i.e. locally.). The model must learn which agents are the adults and which are the children based on its experience seeing the objects with which each is paired. The children all watch one set of shows, and the adults all watch another set. Age, therefore, organizes the sentences into sets. Since age does not appear in the input, however, it must be induced by the model and accessed as each sentence is processed.

Before the model was trained on the corpus of sentences, 50 of the 400 sentences (12.5%) were randomly picked to be set aside as the generalization set. The model was then trained on the remaining sentences. When these sentences were processed correctly (after 90,000 trials), the model was tested for compositional and predictive generalization.

Again, to be compositional, the model should be able to process correctly the sentences on which it had not been trained. The model processed 86% of these sentences correctly. On the remaining 14%, or 7 sentences, the model activated the wrong object more strongly than it activated the correct object. At the point in training when the model was stopped, then, it could compose novel sentences reasonably well.

Semantic generalization should allow the model to predict subsequent information. There are two types of regularities that the model should discover to help make predictions. One is the general regularity that children do children's activities and that adults do adult activities. To test the learning of this regularity we observed which objects the model predicted after the model had processed the agent and action. Given an agent and an action, the model should predict the objects appropriate for that action and an agent of that age. For example, given the partial sentence, "Bobby watched. . .," the model should predict the object to be either He-Man, Smurfs, Mighty Mouse, the Road Runner, or Star Trek. Specifically, this regularity suggests that the model should activate each of these subjects 1/5 or 0.2.

What makes this a generalization task is that some of the sentences in the corpus were set aside and not trained: some agents were never paired with certain objects. For example, the network was never trained on the sentence, "Bobby watched He-Man." To activate each object in the age appropriate set to 0.2, the model must generalize. It must generalize both to discover the complete set of five children's shows, and to associate that set to each of the eight children. This generalization can be tested by observing whether He-Man is activated as a predicted object for "Bobby watched. . . ."

The other type of regularity is specific to each agent. The fact that the training corpus does not contain, "Bobby watched He-Man," is also a regularity that the model can learn. It learns that it should not predict He-Man as an object for "Bobby watched. . . ." Since, for Bobby, there are only four shows that he actually watches, each should be given an activation level of 1/4 or 0.25. The model's tendency to generalize by using the general regularity about



children's viewing habits, therefore, is counteracted by the specific regularity about Bobby's personal viewing habits. The activation of the appropriate/untrained (e.g. He-Man for Bobby) objects should be a compromise between the two competing forces. The appropriate/untrained objects, then, should have an activation somewhat less than 0.2. For the appropriate/trained (e.g. Mighty Mouse for Bobby) objects, the activation level should lie between 0.2 and 0.25, and the activation level for the inappropriate/untrained objects (e.g. the news for Bobby) should be close to 0.

The model's predictions on each of the five actions for four of the people in the corpus were tabulated. The activation values for the model's object predictions were categorized into three groups: the appropriate/trained, the appropriate/untrained, and the inappropriate/untrained. The average activation in each group is shown in Table 2. The means are significantly different from one another.

Three conclusions can be drawn from these data. One, the appropriate/trained objects are activated to the appropriate degree. The model correctly predicts the correct set of objects for the action and the age of the agent. Two, the model generalizes in that it also activates age-correct objects it has not actually seen in that context before. The substantially weaker activation of these appropriate/untrained objects demonstrates the competition between the general and specific regularities. The general regularity activates the objects, but the agent-specific regularities reduce that activation. Three, the set of inappropriate/untrained objects for each action is turned off.

The difference in activation between the appropriate/untrained and inappropriate/untrained objects understates their difference in the network. As the activation of a unit approaches 1 or 0, the non-linearity of the activation function requires that exponentially more activation be added to move closer to 1 or 0. A large change in the net input, therefore, would be required to reduce the activation level of an appropriate/untrained object to that of an inappropriate/untrained object. Consequently, the two types of objects are substantially different, and the general regularity is having a significant impact on prediction.

Interestingly, there is a trade-off during training between compositional generalization and predictive generalization. During training the model improves its ability to compose novel sentences, but loses the ability to make general predictions. The general predictions are lost as the model learns the

Table 2
Semantic prediction

| Appropriateness | Training | |
|---|---|---|
| | Trained | Untrained |
| Appropriate | 0.210 | 0.036 |
| Inappropriate | — | 0.006 |

specific regularities of the training corpus. Compositionality is gained as the input constituents become lexeme-like. As the network trains, it slowly learns which parts of the input are responsible for which parts of the output. The network slowly hones the meanings of the input constituents until they are essentially represented as lexemes.

The model's training procedure encourages predictive generalization for both types of regularities by explicitly requiring the model to predict the object after processing the agent and action. Recall that the model must answer questions about the entire sentence after processing each constituent. The model learns that all of the untrained objects, both appropriate and inappropriate, should not be predicted because they never occur.

It may seem that the model diverges from human behavior as it reduces the influence of the general regularity in favor of the person-specific regularities. It must be remembered, though, that the model is trained on a set of only 16 people. If there were many more people, the person-specific regularities would receive less practice and the general regularity would receive more practice. This change would make the age generalization much stronger.

The model can clearly learn and productively apply regularities from its training corpus to novel sentences. However, the degree to which the model composes novel sentences, 97% in the syntactic corpus, but only 86% in the semantic corpus, is far from the degree we expect from people. Again, this performance may be due to the size and content of the corpus. The effects of these factors on generalization are not well understood. Our simulation should be taken only as an indication that some degree of compositionality can be acquired.

As demonstrated by learning the rule for the passive voice construction, it can learn syntactic regularities, and as demonstrated by learning the meaning of each word and the hidden-feature age, it can learn semantic regularities. Generalization based on these regularities takes two forms. When the regularity consists of input/output pairings, generalization leads to compositionality. Input/output pairings are like lexemes in that the specific input constrains a specific part of the output. There are few, if any, constraints on other parts of the output. Lexemes will compose in novel combinations easily because each one makes a separate and independent contribution to the interpretation.

When the regularity consists of input/input pairings, generalization can lead to prediction. For input/input pairings, one part of the input affects multiple parts of the output. Some of the parts may be future constituents, so the input, in effect, predicts future inputs. These input/input pairings may not compose easily. The parts may make mutually contradictory predictions. For example in, "Bobby watched the news," "Bobby" predicts that *the news* will not be watched, and "the news" predicts that *Bobby*, a child, will not be watching. These contradictory predictions will create conflict in the sentence gestalt and make the interpretation hard to represent.



Additionally, a novel pairing may not compose easily because the constraints specific to a pair may not have been learned. In other words, there may be constraints, in the environment, that pertain to a pair of input constituents. If the model has never experienced this pair, it cannot know these constraints, and they will not affect the interpretation.

### 3.8. Variable syntactic frames

The model is able to learn and use syntactic information. It can correctly apply syntactic information to assign thematic roles in sentences with reversible verbs. But how complex syntax can the model learn? Since the model can only represent simple sentences in the output layer, it cannot learn or use the complex syntax involved in sentences with embedded clauses. The syntax of sentences involving "gave," however, is relatively complex because of the variability in the location of roles in the sentence. Because it need only involve simple sentences, it is representable in the output layer, and because it is representable, it can be used as the target for error correction. The question is, will the model be able to learn?

We created a corpus consisting of the different legal constructions of "gave."

The busdriver gave the rose to the teacher.
The busdriver gave the teacher the rose.
The teacher was given the rose by the busdriver.
The rose was given to the teacher by the busdriver.
The rose was given by the busdriver to the teacher.

The corpus consisted of 56 events of this type. Each event was equally frequent so that there were no semantic regularities for the model to detect and use to help it assign the agent and recipient thematic roles. The model was able to master this corpus. It learned to correctly assign each of the thematic roles in the event described by the sentence.

There are still many more phenomena on which the model has not been trained or tested. The general point, though, is that the model appears to have the capability to learn and use syntactic constraints productively. Where its limits are is presently unknown.

## 4. Discussion

The SG model has been quite successful in meeting the goals that we set out for it, but it is of course far from being the final word on sentence comprehension. Here we briefly review the model's accomplishments. Following this review, we consider some of its limitations and how they might be addressed by further work.

### 4.1. Accomplishments of the model

One of the principle successes of the SG model is that it correctly assigns constituents to thematic roles based on syntactic and semantic constraints. The syntactic constraints are more difficult for the model to master than the semantic constraints even though we have provided explicit cues to the syntax, in the form of the surface location of the constituents, in the input. The model does, however, come to master these constraints as they are exemplified in the corpus of training sentences. Though syntactic constraints can be significantly more subtle than those our model has faced thus far, those it has faced are fairly difficult. To correctly handle active and passive sentences, the model must map surface constituents onto different roles depending on the presence of various surface cues elsewhere in the sentence.

The model also exhibits considerable capacity to use context to disambiguate meanings and to instantiate vague terms in contextually appropriate ways. Indeed, it is probably most appropriate to view the model as treating each constituent in a sentence as a clue or set of clues that constrain the overall event description, rather than as treating each constituent as a lexical item with a particular meaning. Although each clue may provide stronger constraints on some aspects of the event description than on others, it is simply not the case that the meaning associated with the part of the event designated by each constituent is conveyed by only that constituent itself.

The model likewise infers unspecified arguments roughly to the extent that they can be reliably predicted from the context. Here we see very clearly that constituents of an event description can be cued without being specifically designated by any constituent of the sentence. These inferences are graded to reflect the degree to which they are appropriate given the set of clues provided. The drawing of these inferences is also completely intrinsic to the basic comprehension process: no special separate inference processes must be spawned to make inferences, they simply occur implicitly as the constituents of the sentence are processed.

The model demonstrates the capacity to update its representation as each new constituent is encountered. Our demonstration of this aspect of the model's performance is somewhat informal; nevertheless, its capabilities seem impressive. As each constituent is encountered, the interpretation of all aspects of the event description is subject to change. If we revert to thinking in terms of meanings of particular constituents, both prior and subsequent context can influence the interpretation of each constituent. Unlike most conventional sentence processing models, the ability to exploit subsequent context is again an intrinsic part of the process of interpreting each new constituent. There is no backtracking; rather, the representation of the sentence is simply updated to reflect the constraints imposed by each constituent as it is encountered.

While avoiding backtracking, the model also avoids the computational



explosion of computing each possible interpretation of a sentence as it encounters ambiguous words and thematic role assignments. A simpler model helps explain how the SG model avoids these dual pitfalls. Kawamoto [16] describes an auto-associative model of lexical access. In his model, patterns of activation represent a word and its meaning. For each ambiguous word, there are two patterns, each representing the association of the word with one of its meanings. Positive weights interconnect the units representing a pattern, and negative weights interconnect units between patterns.

When an ambiguous word is processed, it initially activates semantic units representing both meanings. The resulting pattern of activation is a combination of the semantic features of both meanings, so the bindings among the features of a meaning are lost in the activation pattern. These bindings, however are preserved in the weights, and the model will settle into one interpretation of the word or the other. One can think of the alternative interpretations as minima in an energy landscape. The initial pattern of activation falls on the high energy ridge between the minima. The settling process moves the pattern of activation down one side of the ridge to one of the minima.

The SG model does not settle, but the idea is similar. When there is not sufficient information to resolve an ambiguity, the sentence interpretations may become conflated in the pattern of activation over the sentence gestalt and in the output responses to probes. The bindings within an interpretation, however, are preserved in the weights. When sufficient new information for disambiguation is provided, the sentence gestalt computes a single interpretation with correct thematic role and semantic feature bindings. For example, given "The adult ate," and with *agent* as the probe, the model partially activates *busdriver*, *teacher*, *male*, and *female*. Which is male and which is female is lost in the activation pattern over the output layer. (It is not clear to what extent the bindings are lost in the activation pattern over the sentence gestalt. In general, the representation over the sentence gestalt is an area for further inquiry.). After the model is further given ". . . the steak with gusto," it is clear that the agent is the busdriver. When probed for the agent, the model unambiguously activates *busdriver* in the output layer.

For the unambiguous sentences in the corpus, the model predominately achieves the correct bindings and interpretations. For the ambiguous sentences, the model conflates, in the output layer, the patterns for each possible role filler for the role being probed. Theoretically, the model should set the activations to match the conditional probabilities for the aspects of the event that remain underspecified. Instead, these activations tend to vacillate based on recent, related training trials. This vacillation toward alternate interpretations is reminiscent of the frequent finding that humans generally do not notice the ambiguity of sentences. Instead, they generally settle for one interpretation or

the other, unless their attention is explicitly drawn to the ambiguity. In sum, the long-term, average probabilities of picking particular interpretations may reflect the statistical properties of the environment, while the moment to moment fluctuations of interpretations reflect recent experience.

The gradual, incremental learning capabilities of the network underlie its ability to solve the bootstrapping problem, that is, to learn simultaneously about both the syntax and semantics of constituents. The problem of learning syntax and semantics is central for developmental psycholinguistics. Naigles, Gleitman and Gleitman [25] state that learning syntax and semantics using only statistical information seems impossible because, "at a minimum, it would require such extensive storage and manipulation of contingently categorized event/conversation pairs as to be unrealistic." Yet it is exactly by using such information that our model solves the problem. The model learns the syntax and semantics of the training corpus simultaneously. Across training trials, the model gradually learns which aspects of the event description each constituent of the input constrains and in what ways it constrains these aspects.

The problem of discovering which event in the world a sentence describes when multiple events are present would be handled in a similar way, though we have not modeled it. Again, the aspects of the world that the sentence actually describes would be discovered gradually over repeated trials, while those aspects that spuriously co-occur with these described aspects would wash out. For both the bootstrapping and the ambiguous reference problem, then, our model takes a gradual, statistical approach. We do not want to overstate the case here, since the child learning a language confronts a considerably more complex version of these problems than our model does. Our sentences are pre-segmented into constituents, are very simple in structure, and are much fewer in number than the sentences a child would hear. However, the results demonstrate that the bootstrapping and ambiguous reference problems might ultimately be overcome by an extension of the present approach.

Many of the accomplishments of the SG model are shared by predecessors. Cottrell, [5] Cottrell and Small [6], Waltz and Pollack [35], and McClelland and Kawamoto [21] have all demonstrated the use of syntactic and semantic constraints in role assignment and meaning disambiguation. Of these, the first two embodied the immediate update principle, but did not learn, while the third learned in a limited way, and had a fixed set of input slots.

The greater learning capability of our model allows it to find connection strengths that solve the constraints embodied in the corpus without requiring the modeler to induce these constraints and without the modeler trying to build them in by hand. It also allows the model to construct its own representations in the sentence gestalt, and this ability allows these representations to be considerably more compact than in other cases.

Some previous models have used conjunctive representations in which



role/filler pairs are explicitly represented by units pre-assigned to represent either specific role/filler pairs [5, 6, 35] or particular combinations of role features and filler features [21]. Particularly, when such representations are extended so that triples, rather than simply pairs, can be represented [30, 33], these networks can become intractably large even with small vocabularies. The present model avoids intractable size by learning to use its representational capacity sparingly to represent just those role/filler pairings that are consistent with its experience. This ability prevents the model from being able to represent totally arbitrary events: its representational capacities are strongly constrained by the range of its experience. In this regard the model seems similar to humans: it is widely known that human comprehension is strongly influenced by experience [2, 4].

The major simulation reported here contained explicit surface role markings in the input. These markings were designed to make the learning of the syntactic information easier. Additional simulations, though, showed that the network could learn the syntactic information from only the temporal sequences of constituents. In a different task, where the network must attempt to anticipate the next input, there have been several demonstrations that networks can learn to keep track of parse position, at least for small finite-state grammars [7, 29]. Extracting information from the temporal order of input sequences, then, is a function generally within the computational limits of recurrent networks.

Finally, the model is able to generalize the processing knowledge it has learned and apply it to novel sentences. Generalization can come in two varieties: compositional and predictive generalization. Compositional generalization occurs when the model has learned the constraints on sentence interpretation contributed by each element of the sentence, including both syntax and semantics, and has learned how to combine that information in novel sequences. The cluster analysis of the input weights confims that the network is learning the semantic constraints imposed by constituents. It seems likely that a considerable part of the specification of these constraints might be derivable by the network from experience on a subset of the possible contexts where a word can occur. The interpretation acquired in these experiences would then cause the new word to behave like other similar words in contexts in which it was not trained. Illustrations that backpropagation networks can generalize in this way are provided by Hinton [12], Taraban, McDonald and MacWhinney [31], and Rumelhart [27]. In both the syntactic and the semantic generalization corpora, the model demonstrates its ability to learn this information and compose it.

The second variety of generalization is predictive generalization. It occurs when the model can use a regularity to predict upcoming sentence constituents. In the semantic generalization experiment, the model learned regularities about the age of agents in the corpus. The model then used these regularities to predict appropriate objects.

### 4.2. Deficiencies and limitations of the model

The model has several limitations and a few obvious deficiencies. The model only addresses a limited number of language phenomena. It does not address quantification, reference and co-reference, coordinate constructions, or many other phenomena. Perhaps the most important limitation is the limitation on the complexity of the sentences, and of the events that they describe. In general, it is necessary to characterize the roles and fillers of sentences with respect to their superordinate constituents. Similarly in complex events, there may be more than one actor, each performing an action in a different sub-event of the overall event or action. Representing these structures requires head/role/filler triples instead of simple role/filler pairs.

One solution is to train the model using triples rather than pairs as the sentence and event constituents. The difficulty lies in specifying the non-sentence members of the triples. These non-sentence members would stand for entire structures. Thus they would be very much like the patterns that we are currently using as sentence gestalts. It would be desirable to have the learning procedure induce these representations, but this is a bootstrapping problem that we have not yet attempted to solve.

Another limitation of the model is the use of local representations both for concepts and for roles. The present model used predominantly local representations of concept meanings only for convenience; in reality we would suppose that the conceptual representations underlying events would be represented by distributed patterns [14]. This kind of representation would have several advantages. Context has the capability not only of selecting among highly distinct meanings such as between flying bats and baseball bats, but also, we believe, of shading meanings, emphasizing certain features and altering properties slightly as a function of context [21]. Both of these phenomena are easily captured if we view the representation of a concept as a distributed pattern.

Similarly, there are several problems with the concept of role which are solved if distributed representations are used. It is often difficult to determine whether two roles are the same, and it is very difficult to decide exactly how many different roles there are. If roles were represented as distributed patterns, these issues would simply fall by the wayside. In earlier work [21], it was necessary to invent distributed representations for concepts, but recently a number of researchers have shown that such representations can be learned [12, 24, 28]. The procedure should also apply to distributed representations of roles.

A final limitation is the small size of the corpus used in training the model. Given the length of time required for training, one might be somewhat pessimistic about the possibility that a network of this kind could master a substantial corpus. However, it should be noted that the extent to which



learning time grows with corpus size is extremely hard to predict for connectionist models, and is highly problem dependent. For some problems (e.g. parity), learning time per pattern increases more than linearly with the number of training patterns [32], while for other problems (e.g. negation), learning time per pattern actually can decrease as the number of patterns increases [27].

Where the current problem falls on this continuum is not yet known. A comparison of the learning times between the general corpus and the syntactic corpus used in the generalization experiments, however, is suggestive. The network required 630,000 trials to learn the 120 events in the general corpus (330,000 trials to learn all but the most irregular events). On the other hand, the network required only 100,000 trials to learn the 2000 events of the syntactic corpus. The syntactic corpus, of course, is extremely regular, and the regularities are compositional. Given the model's good generalization results on the syntactic corpus, it is possible that the model will scale well to very large corpora if their regularities are composable.

One final deficiency of the model is its tendency to activate fillers for roles that do not apply to a particular frame. This tendency could perhaps be overcome by explicit training that there should be no output for a particular role, but this seems inelegant and impractical, especially if we are correct in believing that the set of roles is open-ended. The absence of roles seems somehow implicit in events, rather than explicitly noted. Perhaps event representations that preserved more detail of the real-world event would provide the relevant implicit constraints.

## 5. Conclusion

The SG model represents another step in what will surely be a long series of explorations of connectionist models of language processing. The model is an advance in our view, but there is still a very long way to go. The next step is to find ways to extend the approach to more complex structures and more extensive corpora, while increasing the rate of learning.

## Appendix A. Input and Output Representations

### Input

*surface locations*:
pre-verbal, verbal, post-verbal-1, post-verbal-n

*words*
consumed, ate, drank, stirred, spread, kissed, gave, hit, shot, threw, drove, shed, rose

someone, adult, child, dog, busdriver, teacher, schoolgirl, pitcher, spot
something, food, steak, soup, ice cream, crackers, jelly, iced-tea, kool-aid
utensil, spoon, knife, finger, gun
place, kitchen, living-room, park, bat, ball, bus, fur
gusto, pleasure, daintiness
with, in, to, by
was

**Output**

*roles*:
agent, action, patient, instrument, co-agent, co-patient, location, adverb, recipient

*actions and concepts*:
ate, drank, stirred, spread, kissed, gave, hit, shot, threw(tossed), threw(hosted), drove(transported), drove(motivated), shed(verb), rose(verb)
busdriver, teacher, schoolgirl, pitcher(person), spot
steak, soup, ice cream, crackers, jelly, iced-tea, kool-aid
spoon, knife, finger, gun
kitchen, living-room, shed(noun), park
rose(noun), bat(animal), bat(baseball), ball(sphere), ball(party), bus, pitcher(container), fur
gusto, pleasure, daintiness

*action features*:
consumed, passive

*concept features*:
person, adult, child, dog, male, female
thing, food, utensil
place, in-doors, out-doors

## Appendix B. Sample Sentence-Frame

**Hit**

In the sentence-frame below, superior numbers 1, 2, 3, 4 have the following meaning:

[1] Include a role in the input with this probability.
[2] Choose this filler with this probability.
[3] Choose this word with this probability.
[4] The word appears in this prepositional phrase.



agent 100[1]
  25[2] busdriver 70[3] adult 20 person 10
    verb 100
      100 hit 100
    patient 100
      25 shed-n 80 something 20
        instrument 50
          100 bus 80 something 20 with[4]
      40 ball-s 80 something 20
        location 50
          100 park 100 in
        instrument 50
          100 bat-b 80 something 20 with
      10 bat-a 80 something 20
        location 50
          100 shed-n 100 in
        instrument 50
          100 bat-b 80 something 20 with
      25 pitcher-p 70 child 20 person 10
        location 50
          100 park 100 in
        instrument 50
          100 ball-s 80 something 20 with
  25 teacher 70 adult 20 person 10
    verb 100
      100 hit 100
    patient 100
      34 pitcher-c 80 something 20
        location 50
          100 kitchen 100 in
        instrument 50
          100 spoon 80 something 20 with
      33 pitcher-p 70 child 20 person 10
        location 50
          100 living-room 100 in
        instrument 50
          100 pitcher-c 80 something 20 with
      33 schoolgirl 70 child 20 person 10
        location 50
          100 kitchen 100 in
        instrument 50
          100 spoon 80 something 20 with
  25 pitcher-p 70 child 20 person 10

```
verb 100
  100 hit 100
patient 100
  40 ball-s 80 something 20
    location 50
      100 park 100 in
    instrument 50
      100 bat-b 80 something 20 with
  10 bat-a 80 something 20
    location 50
      100 shed-n 100 in
    instrument 50
      100 bat-b 80 something 20 with
  25 bus 80 something 20
    location 50
      100 park 100 in
    instrument 50
      100 ball-s 80 something 20 with
  25 busdriver 70 adult 20 person 10
    location 50
      100 park 100 in
    instrument 50
      100 ball-s 80 something 20 with
25 schoolgirl 70 child 20 person 10
  verb 100
    100 hit 100
  patient 100
    34 pitcher-c 80 something 20
      location 50
        100 kitchen 100 in
      instrument 50
        100 spoon 80 something 20 with
    33 spot 80 dog 20
      location 50
        100 kitchen 100 in
      instrument 50
        100 spoon 80 something 20 with
    33 teacher 70 adult 20 person 10
      location 50
        100 kitchen 100 in
      instrument 50
        100 spoon 80 something 20 with
```



## REFERENCES


1. R.C. Anderson and A. Ortony, On putting apples into bottles: A problem of polysemy, *Cognitive Psychol.* **7** (1975) 167–180.
2. F.C. Bartlett, *Remembering: An Experimental and Social Study* (Cambridge University Press, Cambridge, 1932).
3. P.A. Carpenter and M.A. Just, Reading comprehension as the eyes see it, in: M.A. Just and P.A. Carpenter, eds., *Cognitive Processes in Comprehension* (Erlbaum, Hillsdale, NJ, 1977).
4. W.G. Chase and H.A. Simon, Perception in chess, *Cognitive Psychol.* **4** (1973) 55–81.
5. G.W. Cottrell, A connectionist approach to word sense disambiguation, Dissertation, Computer Science Department, University of Rochester, NY (1985).
6. G.W. Cottrell and A.L. Small, A connectionist scheme for modeling word sense disambiguation, *Cognition and Brain Theory* **6** (1983) 89–120.
7. J.L. Elman, Finding structure in time, CRL Tech. Rept. 8801, Center for Research in Language, University of California, San Diego, La Jolla, CA (1988).
8. T.D. Erickson and M.E. Mattson, From words to meaning: A semantic illusion, *J. Verbal Learn. Verbal Behav.* **20** (1981) 540–551.
9. C.J. Fillmore, The case for case, in: E. Bach and R.T. Harms, eds., *Universals in Linguistic Theory* (Holt, New York, 1968).
10. L.R. Gleitman and E. Wanner, Language acquisition: The state of the state of the art, in: E. Wanner and L.R. Gleitman, eds., *Language Acquisition: The State of the Art* (Cambridge University Press, Cambridge, MA, 1982).
11. G.E. Hinton, Implementing semantic networks in parallel hardware, in: G.E. Hinton and J.A. Anderson, eds., *Parallel Models of Associative Memory* (Erlbaum, Hillsdale, NJ, 1981).
12. G.E. Hinton, Learning distributed representations of concepts, in: *Proceedings Eighth Annual Conference of the Cognitive Science Society*, Amherst, MA (1986).
13. G.E. Hinton, Connectionist learning procedures, Tech. Rept. CMU-CS-87-115, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA (1987).
14. G.E. Hinton, J.L. McClelland and D.E. Rumelhart, Distributed representations, in: D.E. Rumelhart, J.L. McClelland and the PDP Research Group, eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* **1**: *Foundations* (MIT Press, Cambridge, MA, 1986).
15. M.I. Jordan, Attractor dynamics and parallelism in a connectionist sequential machine, in: *Proceedings Eighth Annual Conference of the Cognitive Science Society*, Amherst, MA (1986).
16. A.H. Kawamoto, Distributed representations of ambiguous words and their resolution in a connectionist network, in: S.L. Small, G.W. Cottrell, and M.K. Tanenhaus, eds., *Lexical Ambiguity Resolution: Perspectives from Psycholinguistics, Neuropsychology, and Artificial Intelligence* (Morgan Kaufmann, San Mateo, CA, 1988).
17. B. MacWhinney, E. Bates and R. Kliegl, Cue validity and sentence interpretation in English, German, and Italian, *J. Verbal Learn. and Verbal Behav.* **23** (1984) 127–150.
18. M.P. Marcus, *A Theory of Syntactic Recognition for Natural Language* (MIT Press, Cambridge, MA, 1980).
19. W. Marslen-Wilson and L.K. Tyler, The temporal structure of spoken language understanding, *Cognition* **8** (1980) 1–71.
20. J.L. McClelland and J.L. Elman, Interactive processes in speech perception: The TRACE model, in: J.L. McClelland, D.E. Rumelhart and the PDP Research Group, eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* **2**: *Applications* (MIT Press, Cambridge, MA, 1986).
21. J.L. McClelland and A.H. Kawamoto, Mechanisms of sentence processing: Assigning roles to constituents, in: J.L. McClelland, D.E. Rumelhart and the PDP Research Group, eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* **2**: *Applications* (MIT Press, Cambridge, MA, 1986).

22. J.L. McClelland and D.E. Rumelhart, An interactive activation model of context effects in letter perception: Part 1. An account of basic findings, *Psychol. Rev.* **88** (1981) 375–407.
23. G. McKoon and R. Ratcliff, The comprehension processes and memory structures involved in instrumental inference, *J. Verbal Learn. and Verbal Behav.* **20** (1981) 671–682.
24. R. Miikkulainen and M.G. Dyer, Building distributed representations without microfeatures, Tech. Rept., Artificial Intelligence Laboratory, Computer Science Department, University of California, Los Angeles, CA (1988).
25. L.G. Naigles, H. Gleitman and L.R. Gleitman, Syntactic bootstrapping in verb acquisition: Evidence from comprehension, Tech. Rept., Department of Psychology, University of Pennsylvania, Philadelphia, PA (1987).
26. W.V. Quine, *Word and Object* (Harvard Press, Cambridge, MA, 1960).
27. D.E. Rumelhart, colloquium presented to the Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA (1987).
28. D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning internal representations by error propagation, in: D.E. Rumelhart, J.L. McClelland and the PDP Research Group, eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* **1**: *Foundations* (MIT Press, Cambridge, MA, 1986).
29. D. Servan-Schreiber, A. Cleeremans and J.L. McClelland, Encoding sequential structure in simple recurrent networks, Tech. Rept. CMU-CS-88-183, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA (1988).
30. M.F. St. John and J.L. McClelland, Reconstructive memory for sentences: A PDP approach, *Proceedings Inference: OUIC 86*, University of Ohio, Athens, OH (1987).
31. R. Taraban J. McDonald and B. MacWhinney, Category learning in a connectionist model: Learning to decline the German definite article, in: R. Corrigan, ed., *Milwaukee Conference on Categorization* (Benjamins, Philadelphia, PA, to appear).
32. G. Tesauro, Scaling relationships in back-propagation learning: Dependence on training set size, Tech. Rept., Center for Complex Systems Research, University of Illinois at Urbana-Champaign, Champaign, IL (1987).
33. D.S. Touretzky and S. Geva, A distributed connectionist representation for concept structures, in: *Proceedings Ninth Annual Conference of the Cognitive Science Society*, Seattle, WA (1987).
34. T.A. van Dijk and W. Kintsch, *Strategies of Discourse Comprehension* (Academic Press, Orlando, FL, 1983).
35. D.L. Waltz and J.B. Pollack, Massively parallel parsing: A strongly interactive model of natural language interpretation, *Cognitive Sci.* **9** (1985) 51–74.