ELSEVIER

Cognitive Science 26 (2002) 609–651

COGNITIVE SCIENCE

http://www.elsevier.com/locate/cogsci

Symbolically speaking: a connectionist model of sentence production

Franklin Chang*

Beckman Institute, University of Illinois at Urbana-Champaign, 405 North Mathews Avenue, Urbana, IL 61801, USA

Received 10 September 2001; received in revised form 16 May 2002; accepted 28 May 2002

Abstract

The ability to combine words into novel sentences has been used to argue that humans have symbolic language production abilities. Critiques of connectionist models of language often center on the inability of these models to generalize symbolically (Fodor & Pylyshyn, 1988; Marcus, 1998). To address these issues, a connectionist model of sentence production was developed. The model had variables (role-concept bindings) that were inspired by spatial representations (Landau & Jackendoff, 1993). In order to take advantage of these variables, a novel dual-pathway architecture with event semantics is proposed and shown to be better at symbolic generalization than several variants. This architecture has one pathway for mapping message content to words and a separate pathway that enforces sequencing constraints. Analysis of the model's hidden units demonstrated that the model learned different types of information in each pathway, and that the model's compositional behavior arose from the combination of these two pathways. The model's ability to balance symbolic and statistical behavior in syntax acquisition and to model aphasic double dissociations provided independent support for the dual-pathway architecture.

Keywords: Neural networks; Psychology; Language acquisition; Learning; Cognitive architecture; Computer simulation

An important use of language is to be able to talk about novel events and circumstances. In order to do this, we need the ability to take the words that we know, and combine them in novel ways. Applying knowledge to a new situation involves generalizing that knowledge beyond the context in which it was originally learned. For example, we can use nouns in sentence

* Tel.: +1-217-244-5494; fax: +1-217-244-8371.
E-mail address: fchang@osgood.cogsci.uiuc.edu (F. Chang).

0364-0213/02/$ – see front matter
PII: S0364-0213(02)00079-4

frames that they have never been paired with before. If I teach you the count noun *blicket*, you can produce the sentence *A blicket is a blicket*, even though you have never heard *blicket* used in this manner. This ability to combine words and sentence frames in the absence of previous experience has led some researchers to argue that language requires symbolic capabilities, where knowledge about language is phrased in terms of variables and operations on those variables (Fodor & Pylyshyn, 1988; Marcus, 1998; Pinker & Prince, 1988).

In addition to arguments for symbolic processing, there is research that shows that people are recording the detailed statistical properties of the sentences that they are hearing and producing. One source of evidence for this is the role of frequency in language processing, where frequencies of words and syntactic structures seem to influence the processing of language (Garnsey, Pearlmutter, Myers, & Lotocky, 1997; MacDonald, Pearlmutter, & Seidenberg, 1994). If the statistical regularities are sufficiently rich, then when people encounter novel language sequences, they can use the similarity of the novel sentences to other sentences that they have experienced to process these novel sequences.

Given that the language system seems to require both symbolic and statistical types of knowledge, theories have been developed which use separate mechanisms to implement these two types of processing, and hence these theories have been called dual mechanism theories. One example of this type of theory concerns the processing of the English past-tense. The English past-tense has a regular form (e.g., walk–walked) and several exceptional cases (e.g., run–ran). Pinker and Prince (1988) offer a dual mechanism account in which the regular form is handled by a symbolic mechanism (a rule that uses variables), and exceptional cases are handled by a mechanism that is sensitive to statistical regularities (spreading activation in a lexical network). Some theorists, however, have argued that statistical learning is powerful enough to explain both symbolic and statistical processing using a single mechanism (Plunkett & Juola, 1999; Rumelhart & McClelland, 1986).

There is some evidence that certain classes of connectionist models do not generalize in the same way that people do. For example, Marcus (1998) found that a simple recurrent network (SRN) could learn equivalence relations like *A rose is a rose* or *A tulip is a tulip*, but when given a novel sentence fragment like *a blicket is a . . .*, the SRN could not predict that *blicket* was going to be the next word. Rather, the model activates all the words that it has seen in this sentence position (e.g., *rose*, *tulip*, etc.). When humans experience equivalence sentence like those above, they often infer that the equivalence relation is intended, and that leads them to complete the novel sentence fragment like *a blicket is a . . .* with the word *blicket*. Because SRNs complete this novel fragment with words that it has seen in similar sequences, it seems to be directly representing the sequences that it has experienced during learning. This suggests that it did not develop abstract variable-based frames like *a X is a X*, where *X* is a variable that can be bound to any word. This limitation is important, because SRNs have been used extensively for modeling acquisition of syntactic frames and the use of statistical regularities in language processing (Christiansen & Chater, 1999; Elman, 1990, 1993; Rohde & Plaut, 1999; St. John & McClelland, 1990).

In some sense, the problem with the generalization ability of SRNs reflects a more basic problem with statistical learning. The more that representations are shaped by experience-driven learning, the more difficult it will be to use these representations in novel situations. The overreliance on experience-driven learning can be reduced by incorporating specialized

mechanisms into connectionist models, and thereby yield models with symbolic abilities (Hummel & Holyoak, 1997; Shastri & Ajjanagadde, 1993). But, because language requires that symbolic representations be bound to lexical and structural representations that are specific to a particular language, and these representations incorporate statistical regularities, it is not clear if these specialized mechanisms would integrate with statistical representations in a way that would yield human-like language performance. So, the task for adherents of connectionist models is to figure out how to guide statistical learning so that it can develop representations that operate symbolically to the extent that humans operate symbolically.

Since "symbol processing" is not an overt behavior, definitions of symbolic computation will vary. Most definitions, though, require that the symbol processor have an ability to bind instances to variables, and use these variables in rules or operations (Fodor & Pylyshyn, 1988; Hadley, 2000; Marcus, 1998). Because the rules or operations operate on variables, they can be used when novel elements are bound to the variables. To provide an explicit account of how symbol processing can be instantiated in a statistical learning system, I compare the generalization ability of several models of sentence production. I first describe the messages and the grammars that all the models will be using (Section 1). Then I present the different model architectures that will be compared (Section 2). The first model architecture, the *Prod-SRN*, is a simple extension of connectionist sequencing models to production. The second model architecture, the *Dual-path model*, is a novel model architecture that has features that allow it to generalize symbolically. One feature of this model is that it makes use of spatial representations, which people use to act symbolically on objects in the world, to help the model do symbolic processing in sentence production. To better understand this architecture, two variants on it will be presented: the *No-event-semantics* and *Linked-path* models. The following section will describe the results of simulations of these model architectures (Section 2.4). Next, I examined three specific tests of symbolic generalization, to understand why the architectures differ (Sections 2.5–2.7). The remainder of the article focuses on the Dual-path model, which was the most successful on the generalization tasks. To see how the model represents these tasks, I examine its internal representations (Section 3). Finally, I show that the model's computational properties explain human acquisition and aphasia data. The acquisition of syntactic structures in the model is compared with acquisition in children, to see if the model constrains overgeneralization of verbs to syntactic structures in a way similar to that of children (Section 4). And, the model will be lesioned to see if its architecture is consistent with double dissociations that are found in aphasia (Section 5).

1. Message structure and sentence grammar

Speaking involves mapping from a set of ideas (which will be called the *message*) to a sequence of words (Bock & Levelt, 1994; Garrett, 1988). To learn this mapping, children must be exposed to sentences in situations where they can infer the message. Language researchers assume that children implicitly learn the internal representations that help them to map between the messages and the sentences, and these representations allow them to produce novel sentences (Pinker, 1989). To simulate this language learning process in training the models, I created a set of training sentences sampled from a grammar. The model learns the

Table 1
Example sentences from the grammar

Sentence type	Subtype	Example
Identity		A dog is a dog.
Locative		The cat is near the cafe.
Motion		A dog go near the church.
Intransitive		The cat sleep.
Transitive	Active	The dog chase the cat.
	Passive	The cat is chase by the dog.
Transfer dative	Prepositional	The man give a cup to the woman.
	Double object	The man give the woman a cup.
Benefactive dative	Prepositional	The man bake the cake for the woman.
	Double object	The man bake the woman a cake.
Change-of-state	Locative-patient	The girl fill the cup with water.
Cause–motion	Patient-locative	The girl pour water into the cup.
Spray–load	Locative-patient	The girl spray the wall with water.
	Patient-locative	The girl spray water onto the wall.

rules of the grammar from the limited number of training sentences, and exhibits that knowledge by producing other sentences that have been generated from the grammar. The grammar was designed to enable the testing of several phenomena from the psychological literature on sentence production. Table 1 shows the types of sentences in the model's grammar. The grammar did not include subject–verb agreement or other verb inflections, because the phenomena under examination did not require these morphemes and eliminating them made the model simpler.

When creating a data set for training or testing, a set of messages was first generated. The messages defined only the propositional content of the target sentence, and did not encode the actual surface structure of the sentence. Each message was created by selecting an action and entities that were appropriate to the action. For example, the action EAT was paired with an entity that was living (the eater) and an object which was not living and not a liquid (the object of eating). This representation would then be used to select lexical items that matched the constraints of the action. So, with the action EAT, the eater could be *man* and the object could be *cake*. The participants in an event were classified into one of three event roles: *agent*, *patient*, *goal*. The agent was the cause of the action, the goal was the final location for the object, and the patient was the object in motion or the affected object. The roles did not match exactly the traditional definitions of these roles (see Dowty, 1991, for arguments about why traditional roles do not work), but instead were designed to increase the generalization capabilities of the model (Chang, Dell, Bock, & Griffin, 2000). For example, the distinction between themes and patients was collapsed into the role of *patient*. Location arguments are not always goals, but they were collapsed into that category for the model. The distinctions between the categories that were collapsed together in the model were expressed with verb-specific semantic information.

The model's lexicon was made up of 20 verbs, 22 nouns, 8 prepositions, 2 determiners, 11 adjectives, and an end of sentence marker. Eight of the nouns were animate, and 14 were inanimate. The verb types included dative (*give*, *throw*, *make*, *bake*), transitive (*hit*, *build*, *eat*, *drink*, *surprise*, *scare*), change-of-state (*fill*), spray–load alternation (*spray*, *load*), cause–motion (*put*, *pour*), intransitive (*sleep*, *dance*), motion (*go*, *walk*) and existence (*is*). For the training and testing sets, most verbs had an equal probability of being selected. But because existence and intransitive verbs were easy to learn, their proportion was reduced, to give the other verbs more training (see Appendix A for details).

For training, each message was paired with a particular sentence structure. Very often, natural languages allow a particular meaning to be expressed with several alternative structures (syntactic alternations) as shown in Table 1. For example, active and passive voice sentences (the transitive alternation) have similar meanings, but differ in the order of the noun phrases and their structural properties. Another alternation in the model was the dative alternation, where the prepositional dative and the double object dative can express closely related meanings. This alternation occurred with both transfer datives (e.g., *give*, *throw*) and benefactive datives (e.g., *make*, *bake*). The last alternation, the spray–load alternation, varied the order of the patient and the goal. The generation of sentences was arranged so that 80% of transitive sentences were paired with active voice, and the rest with passives. For datives and spray–load structures, each alternative occurred approximately 50% of the time. To create some extra variability in the structures that were produced, these percentages were modified by the animacy of the arguments in the sentences, so that animate nouns would tend to go before inanimate nouns 70% of the time (in structures that could alternate). The distribution of structures in the grammar vastly oversimplified the real frequencies of these structures in the world, but maintained some of their character within the alternations.

The relationship between meaning and structure in language is not arbitrary. Rather, there are regularities in the way that arguments in a message are expressed in syntactic distributions. It has been argued that the mapping of meaning into form represents a unit of language knowledge, called a *construction*, and constructions are useful in explaining how people use their syntactic knowledge (Goldberg, 1995). An important feature of a construction is that its meaning is not simply a combination of the meaning of its component words, because speakers can generalize words to constructions that they have never been paired with before (e.g., *sneeze* is intransitive, but you can say *I sneezed the napkin across the table* to encode that sneezing was the cause of the motion of the napkin). The meaning of each construction is represented with *event semantics*, which is different from the semantics associated with lexical concepts. Event semantics identify similarities among constructions (Table 2) and thus helped the models generalize from one construction to a related construction. For example, the intransitive motion construction (e.g., *The girl goes to the cafe*) is related to the cause–motion construction (e.g., *The woman put the dog onto the table*), because the *girl* and the *dog* are both undergoing motion. This is represented by having both constructions share the event feature MOTION. The cause–motion construction was also related to the transfer construction (e.g., *The man gives the dog to the girl*), because they shared both the features CAUSE and MOTION. There is evidence that both children and adults are sensitive to these event features in their language knowledge (Fisher, Gleitman, & Gleitman, 1991; Gropen, Pinker, Hollander, Goldberg, & Wilson, 1989; Gropen, Pinker, Hollander, & Goldberg, 1991).

Table 2
Constructions

Sentence type	Event semantics	Verbs
Identity	EXIST	is
Locative		is
Motion	MOTION	**go**, walk
Intransitive		sleep, dance
Transitive	CAUSE AFFECTED	hit, chase, eat, drink
	CAUSE CREATE	**make**, bake
	CAUSE EXPERIENCE	surprise, scare
Transfer dative	CAUSEMOTION TRANSFER	**give**, throw
Benefactive dative	CAUSE AFFECTED TRANSFER	hit, chase, eat, drink
	CAUSE CREATE TRANSFER	**make**, bake
	CAUSE EXPERIENCE TRANSFER	surprise, scare
Change-of-state	CAUSE CHANGE	**fill**
Cause–motion	CAUSEMOTION	**put**, pour
Spray–load	CAUSE CHANGE MOTION	spray, load

As mentioned earlier in the example with the action EAT, the arguments of a verb were constrained to be appropriate for it. To implement this knowledge, each construction was associated with argument constraints. For example, the goals in cause–motion events (e.g., *the cake* in *The woman pushed the car onto the cake*) were allowed to be inanimate, but the goals in transfer dative events were required to be animate (e.g., *the man* in *The woman gave the car to the man*). Another constraint is that adjectives were divided into two classes, those that were restricted to animate arguments (*nice*, *silly*, *funny*, *loud*, *quiet*), and those that were not restricted (*good*, *red*, *blue*, *pretty*, *young*, *old*). These constraints made the sentences that were generated more plausible, but still allowed the grammar to generate some implausible sentences. It is difficult to incorporate all of the world knowledge that is needed to constrain this grammar in the way that human language is constrained. Also, because I will be doing model comparisons, the plausibility or implausibility of the grammar will be the same for all the model types, and so the differences in the models cannot be attributed to these constraints.

In generating the training and testing sentences, the event semantics were used to determine which messages could alternate. In order to alternate, the messages had to be related to two alternative structures by means of these event features (Goldberg, 1995), and each message-structure mapping represented a separate construction. For example, messages with the event features CAUSE, MOTION, TRANSFER could use the double object structure (e.g., *The man give the girl the book*), because this was designated as the default structure for these features. But because this combination of features overlaps with those used in the cause–motion construction (i.e., CAUSE and MOTION), it could also use the prepositional-dative structure (e.g., *The man give the book to the girl*). The spray–load alternation arose because the messages with event features CAUSE, MOTION, and CHANGE were associated with two constructions. The cause–motion construction (licensed by CAUSE and MOTION) selected the structure which put the patient before the goal (e.g., *The man spray the water onto the wall*), while

the change-of-state construction (licensed by CAUSE and CHANGE) was associated with the order that put the goal before the patient (e.g., *The man spray the wall with water*). The passive structure was allowed to alternate with all transitive constructions.

Although event semantics in the intended message can influence the sentence structure that is chosen, speakers can also choose a structure based on other factors. In production studies, people are told to repeat sentences as they hear them, and for the most part, they are able to do this. Here, some verbatim memory of the structure is guiding the choice. But, when doing repetition, people also frequently change the structure of their sentences (Potter & Lombardi, 1990). What this suggests is that there is some information in the message that allows people to control their structure building, but this information is weak enough that sometimes it is overcome by other factors (Bock, 1982). To represent this weak control information, the model made use of the relative activation level of the event semantics. Consider the active–passive alternation. For passives, the AFFECTED feature would be more active than the CAUSE feature, and vice versa for active sentences. For datives, if the TRANSFER feature was more active than the MOTION features, then a double object was produced, otherwise a prepositional dative was produced. For the spray–load alternation, if the feature CHANGE was more active than MOTION, then a locative-patient sentence was produced, otherwise a patient-locative was produced. To set up these differences, I used a prominence parameter (set at 0.8), which controlled the difference in the activation levels for these features. For example, the activation of the feature MOTION was 80% of the activation of the TRANSFER feature if a double object structure was desired.

How do speakers select between alternations in production? Experimental work in sentence production has shown that speakers plan their sentences incrementally, adjusting their structures to fit the words that have come before (Bock, 1982, 1986; Ferreira, 1996; Ferreira & Dell, 2000). To create this ability in the models, the models needed feedback about the previously produced words. Two types of feedback were used: one type that corresponds to the feedback in production and the other that corresponds to the feedback in comprehension. Feedback type was fixed within a sentence, so a sentence could be experienced in either production mode or comprehension mode. Production mode involved passing previously produced words as an input for all the words in a sentence. Correspondingly, comprehension mode involved passing the previous "heard" target words as an input. Comprehension and production modes both attempt to predict the next word in a sequence with a message, but they differ in terms of whether they use an external sequence to help them to do this. Because the model is learning to do production, its production outputs early in training are not very useful for learning the language, so a larger percentage of the training sentences were in comprehension mode (75%) and the rest were in production mode (25%). For testing, the model was always tested in production mode, because we were interested in its production behavior.

The sentence grammar was used to generate 501 training sentences. In order to test the model's ability to generalize, the training sentences had one extra restriction: the word *dog* could never be the goal of the sentence. By testing the model's ability to produce *dog* as the goal of a sentence, even though it was never trained to do so, we can see how well the model generalizes outside of the regularities in the training set. To test overall generalization, a test set was created with 2,000 randomly generated sentences from the grammar. Because the grammar can generate 75,330 possible messages (not including surface form alternations) and the

training set is small, the testing set is mostly made up of novel sentences, and therefore can provide a good picture of the overall accuracy of the model.

2. Symbolic generalization in different architectures

To show that neural networks can exhibit symbolic properties and that their architecture can influence this ability, several architectures will be described and compared. The first architecture (*Prod-SRN* model) is a model that embodies the hypothesis that symbolic generalization is simply due to learning the appropriate statistical representation. This non-symbolic model will be compared to the model that this paper features, called the *Dual-path* model. The Dual-path architecture uses a message based on variables and it sequences these variables by using event semantics. In addition, this model places limits on how sequential information can interact with lexical semantics, effectively creating two pathways in the architecture. To show that the model's behavior is not simply due to the addition of variables, it will be compared to a version of the model that lacks the event semantics (*No-event-semantics* model). To show that the power of the variables also depends on its dual-pathways architecture, the Dual-path model will be compared to a fourth model, the *Linked-path* model, which links up the pathways that the Dual-path model keeps separate. Linking the pathways was expected to diminish the combinatorial abilities of the model.

In addition to the same training and testing sets, the models also shared a few other features. All the models had to have some way of representing the message, and once the message was set, there was no external manipulation of the message during the production of the sentence. All the models were taught to produce words as output, where a single unit represented each word. To increase the models' tendency to choose a single word, the output units employed a soft-max activation function that magnified activation differences (see Appendix A for further details). All the models were trained using back-propagation of error, which is a learning algorithm that computes the difference between the target representation and the model's output and then passes this information back through the network in order to guide weight changes (Rumelhart, Hinton, & Williams, 1986).

2.1. Statistical learning of production: the Prod-SRN model

The Production Simple Recurrent Network (Prod-SRN, Fig. 1) was a SRN (Elman, 1990), which was augmented with a *message* (Dell, Chang, & Griffin, 1999; Jordan, 1986). One part of the network mapped from a representation of the previous word to the next word in the sequence. The output *word* units received inputs from a set of *hidden* units, and the *hidden* units received inputs from the previous word (*cword*, the 'c' indicates that this input is feedback through the comprehension system) and set of *context* units that had a copy of the previous *hidden* unit states.

Because production involves planning a sequence with an intended meaning (as opposed to sequence prediction), the Prod-SRN included a static message. The message was connected to the SRN hidden units, and this allowed the model to use the message to guide the sequence generation. The message representation used binding-by-space (Chang et al., 2000; Dell et al.,

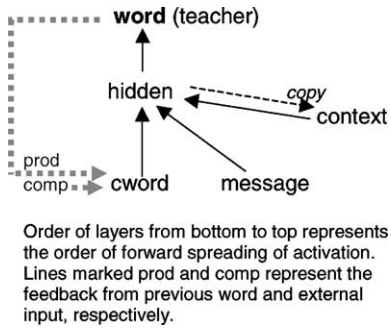

Fig. 1. Prod-SRN model.

1999; McClelland & Kawamoto, 1986; St. John & McClelland, 1990). That is, different event roles were represented by different banks of units. Each bank (or set) of units represented a slot in the message, and there were three role slots (agent, patient, goal) and a slot for the action. Each of the roles had a localist semantic representation: a unit for the meaning of *dog* in the agent slot (e.g., *the dog chased the cat*) and a separate unit for *dog* in the patient slot (e.g., *the cat chased the dog*). Each action was represented by a unique action feature. The event semantics were also included in the message, in the action slot, by giving each event semantic feature its own unit.

Table 3 is an example message for the sentence *A man bake a cake for the cafe*. Because this message has a separate set of semantic features for each slot in the message, the features in each slot are labeled with a number (1 = agent, 2 = patient, 3 = goal) to show that they are different from the same feature in another slot (e.g., CAKE1, CAKE2, CAKE3). The action slot did not overlap with any other slots, so those features are not given the extra number index. In this message, there are *event-semantics* features (CAUSE, CREATE, TRANSFER) and a verb-specific feature (BAKE). Definite articles (*the*) were marked with the slot-specific feature (e.g., DEFINITE3). Indefinite articles (*a*) were not marked (because they cannot occur with mass lexical items, e.g., *coffee*, leaving them unmarked made them depend on the lexical semantic information, since that was the only information that was available).

The output of the model was a localist representation for the words in the lexicon. The *hidden* layer and *context* layer were 50 units each, and the *context* units were initialized to 0.5 at the beginning of each sentence. As mentioned in the section on the training and testing sets, the *cword* representation is set by the target previous word 75% of the time (comprehension

Table 3
Example message (binding-by-space)

Role	Features
Action	BAKECAUSE CREATE TRANSFER
Agent	MAN1
Patient	CAKE2
Goal	CAFE3DEFINITE3

mode) and set to the previously produced word 25% of the time (production mode). Because the model learns production during comprehension, the *cword* units were set to the sum of the previous *word* output and the target previous word. In production, the *cword* units are solely dependent on the model's production output, but in comprehension, the *cword* units were a combination of previous produced word and previous target word. The analog to this in human behavior is that people sometimes mishear what other people say (the *cword* units), because they have filled in their own predicted continuations (the *word* units).

2.2. A symbolic connectionist model: the Dual-path model

The Dual-path model was designed to generalize symbolically, and hence it differed substantially from the Prod-SRN model. In language production, symbolic generalization is exhibited by placing words in novel sentence positions. If you learn a new word, you can use this word in a variety of frames. To get this word-based generalization, the mapping from lexical semantics to word forms should be the same, regardless of where the word occurs in the sentence.

Capturing both lexical and sentence-level aspects of words is similar to a problem in the spatial processing of visual input, where one has to both categorize an object and record its position in a scene (Landau & Jackendoff, 1993). The process of object categorization must remove location-specific information and transform the object to take into account the point of view of the viewer, in order to get an invariant representation that can be used for categorization (Kosslyn, 1980). The process of locating an object, on the other hand, does not need to concern itself with the identity of the object in order to determine the position in space. These two functions have been identified with separate brain structures, the *what* (object) and *where* (location) pathways (Mishkin & Ungerleider, 1982). These two separate representations have to be bound to each other, in order to know which object occurs in which location. The resulting system can recognize known objects in new locations and identify the location of unfamiliar objects. That is, it generalizes well. And it does so because of the separation (and binding) of the object and location information.

Just as the spatial system can generalize in different ways because it has separate *what* and *where* representations, a model of sentence production should be able to generalize well if it represented its message in several separate representations that were linked together. That idea was the basis for the Dual-path model. This architecture had two pathways, one for representing the mapping of object semantics to word forms, and another for representing and mapping objects (and the words that describe them) into appropriate sentence positions (Fig. 2).

The first pathway of the model was the *message–lexical system* (see thick arrows on Fig. 2). This subnetwork was a feed-forward network from the message to the lexicon. The message in this model was represented in weight bindings between a layer of *where* units (thematic) and a layer of *what* units (semantic). By using this type of representation, the same *what* units could represent the meaning of a word, regardless of its event role. The *where* units represented the agent, patient, and goal event roles, and another unit represented action information. The *what* units represented the semantics of words using the same localist representation that was used for the Prod-SRN. Messages in this model were represented by setting the weight between the *where* units and the *what* units to an arbitrary "on" value (see Appendix A for details).

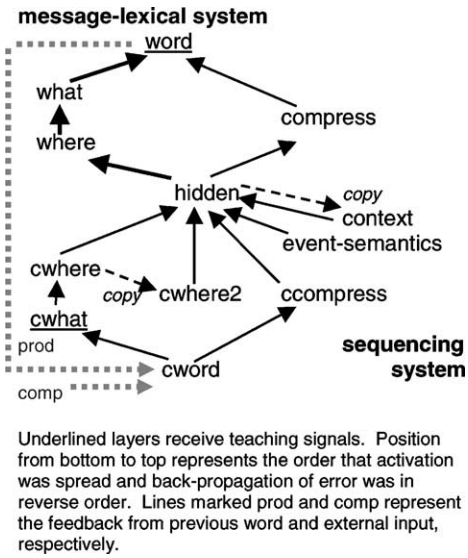

Fig. 2. Dual-path model.

For example, if a dog was the agent, then the agent *where* unit would be connected to the DOG feature in the *what* units. If a dog was the patient, then the patient *where* unit would be connected to the same DOG feature. In this way, we could represent the different roles of dog in these events, while maintaining the common semantics that all dogs share.

The other part of the message–lexical system was the connection between the *what* units and the *word* units. This allowed the model to learn a word label for each meaning. Since there was only one set of *what* units (unlike the Prod-SRN model, which duplicated the analogous units for each event role), learning the mapping of the semantic feature DOG to the word dog allowed the model to generalize this word to other event roles. By using a single semantic/lexical network, this model is more in line with mainstream lexical production theories (Dell, 1986; Levelt, Roelofs, & Meyer, 1999) than the message–lexical mapping of the Prod-SRN model.

The second pathway of the network, the *sequencing system*, was a simple recurrent network with a few other inputs (see thin arrows on Fig. 2). The network mapped from the *cword* units to a *hidden* layer. The *hidden* layer received input from *context* units, which, like the Prod-SRN model, had a history of their previous states as well as a copy of the previous *hidden* unit states. The *hidden* layer then mapped to the same *word* units that are used in the message–lexical system. Like Elman's (1990) model, in between the lexical layers (*cword* and *word*) and the *hidden* units were compression layers (*ccompress*, *compress*), which helped the sequencing network to create generalizations over words, rather than word-specific representations.

The sequencing subnetwork also received input from a reverse version of the message–lexical network ($cword \rightarrow cwhat \rightarrow cwhere \rightarrow hidden$). Without this subsystem, the model would not be able to vary its sentence structures based on the role of the previously produced word. If you said *the cat*, it could be the beginning of the sentence *the cat chased the dog* or *the cat was chased by the dog*. Without knowing what role *cat* plays in your message, you do not know whether to continue the sentence with an active or passive structure. The reverse

message–lexical network tells the sequencing network the role of the last word that the model produced, which allowed it to dynamically adjust the rest of the sentence to match earlier choices. This network mapped *cword* units into *cwhat* units and the *cwhat* units had a variable binding to the *cwhere* units that was set to the analogous reverse *what–where* binding before the initiation of production of a sentence.

In order for the model to use the *cwhat–cwhere* links, it had to learn the mapping between *cword* and *cwhat*. That is, it had to learn the meaning of each word in the comprehension direction. Because the error signal from the *word* units, that is, the produced word, was back-propagated along the weights in the network, error information was weakened as it passes back in the network. The error signal from outputted words was not sufficient to learn the *cword* to *cwhat* mapping in a way that would help the overall learning of production. Therefore, to help these units learn, the *cwhat* units were provided with the previous *what* units' activation as target activations. But since the *what* units activation depend on the links from the *where* units, and initially, the model had not learned to control the *where* units yet, it did not have very good targets to give to the *cwhat* system. What happened was the model bootstrapped word learning, by incrementally learning to comprehend the previously produced semantics. For example, suppose the model was learning a sentence where the agent was a cat. At the beginning of training, the network had random weights. To get an error signal to the *cwhat* units so that the model could learn that *cword* unit *cat* should be linked to the *cwhat* unit CAT, the model needs to activate the production *what* unit CAT by activating the agent unit in the *where* layer. But activation of the *where* layer depended on hidden unit states, and those states in turn depend on *cwhere* information. But slowly, as the model learned to activate the *where* units appropriately in production, the *what* unit activations became more distinctive, and more error was passed back to the *cwhat* units. Intuitively, the model must learn to pick out the role in the message that is associated with the word that it hears. There is evidence that children have similar abilities, in that they can actively guide their attention to elements in a scene to learn the right meanings for words. Children actually go beyond the model in this respect, because in addition to being able to control attention in word learning, they also have sophisticated joint attention abilities to infer intended referents (Baldwin, 1993; Tomasello, 1999).

There were a few other details about the Dual-path architecture that should be mentioned. The *hidden* layer in the Dual-path model was smaller than in the Prod-SRN model (20 units instead of 50 units), because in this architecture, the *hidden* layer did not have the task of mapping all the message elements into words. The *cwhere* units were soft-max units, which forces these units to choose one winner and to reduce the activation of competitors. To help the model to remember what event roles had already been produced, the model also had a set of context units called *cwhere2*. The *cwhere2* units summed the activation from the previous *cwhere* and *cwhere2* states. Because the *cwhere* units were strongly biased to represent the present role of the *cword* input, the *cwhere2* units helped the model to record the history of roles that the model had gone through.

The hidden units also received inputs from a set of units that held the event semantics of the intended construction. These *event-semantics* units helped the sequencing system to create appropriate sequences for that construction. The functionality of the event semantics will be examined by comparing the Dual-path model to the No-event-semantics model, an otherwise identical model that lacked these features.

Table 4
Example message (what–where message and event semantics)

Role	Features
Action	BAKE
Agent	MAN
Patient	CAKE
Goal	CAFE, DEFINITE
Event semantics	CAUSE CREATE TRANSFER

Table 4 shows how the Dual-path model would represent the example message that was used earlier (*A man bake a cake for the cafe*.). Because this model, unlike the Prod-SRN model, only used one set of semantic features, the features were not indexed with a number. The *event-semantics* layer held the construction-specific features.

There were two points where the message–lexical and the sequencing systems interacted. One point was a connection from the *hidden* units of the sequencing system to the *where* units of the message–lexical system. This allowed the model to sequence the *where* units, and that enabled it to produce message-related words in appropriate places. But because the sequencing network did not have access to the message, it tended to develop representations that were independent of the lexical-semantic content of the intended message. That is, its representations tended to be syntactic, as I show later in an analysis of the hidden units. The second point of interaction between these two systems was the *word* units. Here the message–lexical system activated meaning-related possibilities, and the sequencing system activated syntactically-appropriate possibilities. The intersecting activation from these two sources enabled the production of message-appropriate words (message–lexical system) at the proper positions in sentences (sequencing system). The use of separate networks for each mapping is consistent with work in sentence production that showed that lexical-semantic factors and syntactic factors have independent effects on sentence structures (Bock, 1987; Bock, Loebell, & Morey, 1992).

Because of the complexity of the model, it is useful to present an example of how a fully trained model would produce the sentence *A man baked a cake for the cafe*. I will first demonstrate the operation of a trained model in production mode and later comment about the differences that would occur in comprehension mode. Feed-forward connectionist models break down processing into timesteps. In each timestep, activation is propagated forward in the network, and for back-propagation networks, error is back-propagated. Before the first timestep, the message would be set in both the *what–where* and *cwhat–cwhere* links (see Appendix A). In this case, the agent *where* unit would be linked to the MAN semantics, the patient linked to CAKE semantics, and goal linked to CAFE and DEFINITE features (the corresponding reverse links were set in the *cwhat–cwhere* links). Once the message was set, there was no external manipulation of it. The *event-semantics* units would also be set at this time, and since the target sentence is using the benefactive dative construction, the feature CAUSE would be more activated than the feature CREATE, which would be more activated than the feature TRANSFER. Because the model has learned sentences where these *event-semantics* activation values were associated with activations of *where* units that are appropriate for

the benefactive dative structure, the *event-semantics* units are helpful for creating the target order.

After setting the message and *event-semantics* units, production of the first timestep would begin. The *cword* and *context* units needed to be initialized to default values at the beginning of the first timestep, because normally their activation values would come from the activation of other units on previous timesteps. The *cword* units would be set to 0, and the *context* units would be set to 0.5. Activation would then spread from the "bottom" (*cword* units) to the "top" (*word* units) thus identifying a word output. Because the model is in production mode, it must produce the first word without any cword inputs (initialized to 0). The model compares its own output with this target, and the difference (error) is back-propagated to adjust the weights so that the model will be better at producing this word at the beginning of sentences with message like this one.

At the beginning of the second timestep, the *word* output (i.e., *a*) would be copied back to the *cword* units. The activation values of the *what* units at the previous timestep would be used as target values for the *cwhat* units (at this point, this helps the model associate the *a cword* activation with the agent lexical semantics (MAN), but since the model experiences *a* with a variety of different nouns, it doesn't learn a strong connection to any particular noun). The activation of the *cwhere* units would be copied to the *cwhere2* units and summed with the previous activation of those units. The *context* units would receive a copy of the previous *hidden* units states. Activation would spread up to the *word* units (since the model is trained, it should say *man* at this point). The model's output is compared with the comprehended target word *man* and the difference is used to adjust the weight through back-propagation. This process (setting of copied units and input units, spreading activation forward, back-propagation) continues for each word in the sentence. In comprehension mode, the only difference would be that the external comprehended word and the previously produced word would be summed to set the *cword* activation values.

As I mentioned earlier, a primary inspiration for the representation of the message in weights between *what* and *where* units was inspired by the distinction in spatial processing between object and location processing. If visual and other representations are already pre-segmented into these types of representations, and if these representations were bound together with temporary links, then these representations instantiate a type of variable representation, where the location variables can be used to index semantic content. If language makes use of location variables that are instantiated by other systems, then as long as new concepts could be temporarily bound to these variables, then the language system could also make use of these new concepts in constructing its sentences. This arrangement would also allow a connectionist model to have more symbolic abilities, because even if you use statistical learning to develop the sequences that activate the variables, the variables allow novel elements to be incorporated into these sequences. One prediction of this approach is that spatial factors might be influencing language processing, and in the next few paragraphs, I will provide some evidence for this influence.

The idea that spatial factors influence language has a long tradition within certain linguistic theories such as Cognitive Grammar (Lakoff, 1987; Langacker, 1987; Talmy, 1999). In these theories, some syntactic operations are represented as movement through an abstract spatial representation, and therefore these theories are particularly good at explaining why non-motion events make use of motion vocabulary (as in change-of-states such as *His mood went from*

good to bad). The cognitive grammarians argue that non-motion constructions often make use of an abstract spatial-path of a trajector (*mood*) between a source (*good state*) and a goal (*bad state*), and that allows speakers to talk about abstract state changes as movement through a spatial representation. A related claim has been made is that the organization of the spatial system influences the organization of syntactic categories. Landau and Jackendoff (1993) have suggested that the distinction between nouns and prepositions is a direct result of the distinctions in the spatial system between object representations (*what*) and location representations (*where*). The idea that language makes use of perceptual representations is an important part of more general accounts of cognition which argue that most of cognition is inherently modality-specific and involves perceptually represented symbols (Barsalou, 1999).

Developmental psychologists have also argued that spatial representations are important in the development of conceptual representations. Mandler (1992) claimed that children analyze multi-modal perceptual information and redescribe that information internally as image schemas (e.g., path, containment, force). These image schemas represent abstract spatial relationships. In particular, they abstract over the concept fillers that participate in these relationships, and simply treat them as variables. By identifying the relationships among the components of image schemas, children can derive important thematic distinctions like the difference between animacy and agency. Since children must make use of perceptual information to derive the distinctions that are necessary for language, it seems reasonable that perceptual systems would interface in some manner with language.

There is also evidence that the spatial nature of these conceptual representations can influence language acquisition, sometimes in spite of the language input that children receive. An example of this comes from non-conventional uses of *from* by 2- and 3-year-old English speakers. Clark and Carpenter (1989) found that children tended to use *from* to mark agents, in cases where adults would not have marked agents or where a passive would be used with the preposition *by* (e.g., *He's really scared from Tommy* and *I was caught from you before*). They argue that children are collapsing agents and causes into the spatial source category, which is normally marked with the preposition *from* (as in *He drove from home to work*). The use of *from* instead of *by* to mark agents suggests that source is a default category for agents which is later modified to mark agenthood.

In addition to links between the representations in language and spatial processing, there also seems to be evidence that links the spatial system to on-line language production. Griffin and Bock (2000) found that eye movements in picture description were coordinated with the order of elements in sentences that speakers were producing in a way that suggested a tight connection between the two processes. A system that modularizes language and spatial representations by using abstract propositional representations to mediate between them would be unlikely to show this tight connection between eye movements and sentence structure, because in a modular theory, the mediating representations would make it difficult to map backwards from syntactic decisions (e.g., passive structure) to eye movements (e.g., look left). But in a theory with spatial organized messages, where production can be seen as movement of attention over spatial variables, it is easier to understand why syntactic processing and eye movements should be so closely coordinated.

If language and spatial representations are related, how should this relationship be implemented? In the Dual-path model, I have made the assumption that the link between space

and language is limited to the organization of the message. There are two ways that spatial properties have influenced the representation of the message. First is in the separation between object characteristics (*what*) and their relational characteristics (*where*) into separate banks of units with bindings between them. This has the concrete effect of reducing the number of units that are needed to represent the message (the Prod-SRN used 130 units and the Dual-path used 56 units). The second is that the event roles that are instantiated in the *where* units can be thought of as corresponding to components of a spatial "path." The evidence from the linguistics (Jackendoff, 1990; Lakoff, 1987) and developmental literatures (Mandler, 1992) suggests that the distinctions present in thematic roles (agent, patient, experiencer, etc.) can arise from elaborations of spatial roles like source (start of event), theme (object in center of attention), and goal (end of event) that represent a path for an event (see Regier, 1995, for a connectionist model that makes use of a path representation to model the cross-linguistic acquisition of preposition use). This idea is implemented in the model by collapsing thematic-roles distinctions into three spatial roles. In the description of the message representations, I labeled these roles *agent*, *patient*, *goal*, so that they would map onto the appropriate label for most of the sentences in the grammar. But because the *agent* slot collapsed agents and causes, it could also be labeled as an abstract *source* role. Likewise, the *patient* slot collapsed patients, themes, and experiencers, it could be also be relabeled as an abstract *theme* or *object* role. And the *goal* slot represents the roles of goals, recipients, and locations. The distinction among the different thematic roles that used a single spatial role was represented in the event semantic features that were associated with these units (e.g., experiencers had the feature EXPERIENCE, which distinguished it from other patients). As with the what–where distinction, the spatial-path approach to thematic roles also had the effect of reducing the number of *where* units needed to represent the message.

While none of these linguistic, developmental, and processing findings is definitive, there is a growing consensus that language and space are interrelated (see Bloom, Peterson, Nadel, & Garrett, 1999, for a recent summary of this issue). This consensus suggests that we should prefer models that provide a means of linking language and space over those that make no such link or make it very indirectly. The *what–where* representation in the Dual-path model reflects this preference.

2.3. Two alternative architectures: the No-event-semantics and Linked-path models

The existence of variables in the Dual-path model should lead to symbolic abilities in the model. But what is often not recognized by advocates of symbolic theories is that variables require a lot of information to control their use. Having a variable called AGENT does not tell you that agents can occur in the subject position in active English sentences and in a *by*-phrase in English passives (or that they are marked by the particle *ni* in Japanese passives). Rather, language users must learn to use variables in a way that is appropriate to a particular language. The Dual-path model has two characteristics that constrain how variables are used. One is the event semantics, which provide the sequencing system with information about the type of structure that will convey all of the variables in the message. The other is the architecture of the model, which allows the model to ignore the content of the variables when it decides how to use them. To show that these factors influence symbolic generalization, the Dual-path model

will be compared to two models: one that lacks the *event-semantics* units (*No-event-semantics* model) and another that violates the architecture of the Dual-path model (*Linked-path* model).

First, consider the No-event-semantics model. It was designed to be identical to the Dual-path model, except the *event-semantics* units were disabled. Disabling these units prevents the sequencing system from getting information about the intended message, and this makes its representations more syntactic. The sequencing system can only make predictions based on the lexical items that it has produced or comprehended previously. For example, if the *Dual-path* model has the *event-semantics* feature TRANSFER activated, then the model can use that to restrict sentential subjects to animate nouns, since dative subjects in the grammar tended to be animate. But without that information, the model will tend to activate all nouns, because over all the constructions in the grammar, the subject of the sentence could be animate or inanimate. So, a model without event semantics should learn syntactic structures and link them to variables in the message–lexical system. Therefore, the No-event-semantics model implements the idea that syntactic rules and variables are all that a system needs for symbolic generalization. If the Dual-path network is better than the No-event-semantics model, then that suggests that variables are not that useful without information that guides their use (in this case event semantics).

The comparison between the Dual-path and the Linked-path model is an attempt to show that symbolic capabilities are not necessarily associated with the most complex models. The Dual-path model has variables and a special architecture, so it might generalize better than a simpler model like the Prod-SRN model. While it is generally true that simpler symbolic models have fewer capabilities than complex models, connectionist models that use learning to develop their internal representations tend to be opportunistic in their use of information, and therefore more complicated models have more *inappropriate* ways to learn to represent a task.

To demonstrate this, the Dual-path model will be compared with an identical model that has a link between the *what* units and the *hidden* units. This removes the separation between the pathways. This Linked-path model can use these *what–hidden* weights to make use of message information in the sequencing system, and should therefore develop representations that are more optimal for the particular sentences in the training set. This optimization would be expected to reduce the ability to generalize symbolically. If the Dual-path model generalizes better than the Linked-path model, it can be concluded that the separation of the pathways plays an important role in the acquisition of production skills.

2.4. Model comparison experiments

To summarize, four different model architectures were compared (Prod-SRN, Dual-path, No-event-semantics, Linked-path). The Prod-SRN used the binding-by-space message, while the others used the what–where message. The No-event-semantics model was the same as the Dual-path model, except the sequencing system did not have the *event-semantics* units. The Linked-path model was the same as the Dual-path model, except the sequencing system had access to the lexical semantic content of the message through the *what–hidden* links.

Four different training sets (501 sentences each) were created using different random seeds. For each of these sets, the Prod-SRN, Dual-path, No-event-semantics, and Linked-path models were trained for 4,000 epochs. This amount of training resulted in good accuracy within that

amount of time. On analogy with human subjects where different people experience different sentences in their lifetime, the label *model subject* will be used to refer to differences that are due to a particular training set. So, each model type had four model subjects, yielding a total of 16 models. Model weights were initialized to random values between −1 and 1. Every 200 epochs during training, each model was tested on its own training set and a set of 2,000 randomly generated test sentences that was the same for all model subjects. A sentence was accurately produced if the most activated output word (whose activation was higher than the threshold 0.5) matched the target output word for every position in the sentence. The dependent measure in the analyses was the percentage of sentences that were accurately produced in each of the sets.

Averaged over all the model subjects, all of the model types achieved higher than 98% accuracy on the training set by the end of training (Fig. 3). They differed somewhat in the time that it took to reach achieve that accuracy level (Prod-SRN reached it after 3,000 epochs, No-event-semantics reached it after 3,600 epochs, Linked-path reached it after 1,200 epochs, and the Dual-path reached it after 1,400 epochs), but their final accuracy level at 4,000 epochs shows that the architectures ultimately did not differ in their ability to represent the knowledge needed to produce the sentences in the training set.

To test overall generalization, I looked at the accuracy on the set of 2,000 test sentences generated randomly from the grammar (Fig. 4). On these test sentences, the differences among the architectures were evident. The Prod-SRN model never generalized very well. Even as the training accuracy reaches 99%, the testing accuracy never climbs above 13%. The No-event-semantics model did better, reaching a final accuracy of about 52%. The Dual-path and Linked-path models jumped above 70% after 1,200 epochs (as the models were reaching the maximum accuracy on the training set). Here the two diverge, and the Dual-path model reached 79% while the Linked-path model fell to 68% accuracy. A repeated measures analysis of variance (ANOVA) was performed on the accuracy at epoch 4,000 for all four model types with training set as the random factor. Model type was significant [$F(3, 9) = 63.9$, $p < .0001$]. Pairwise

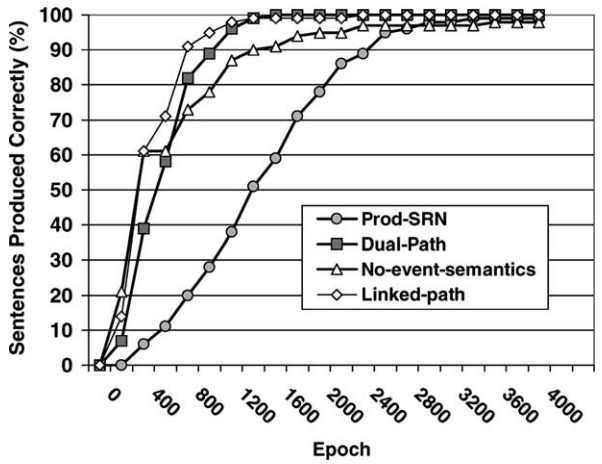

Fig. 3. Average training set accuracy.

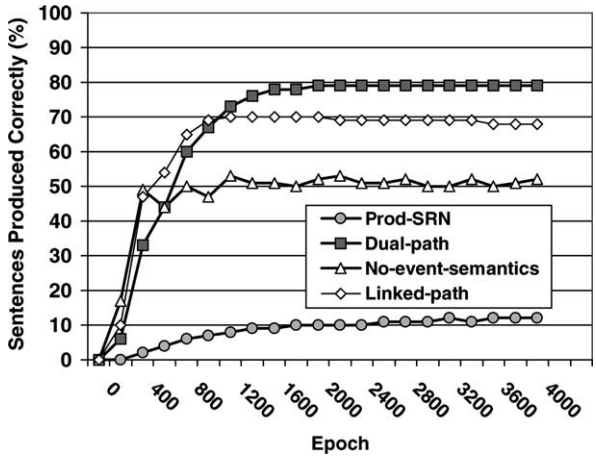

Fig. 4. Average testing set accuracy.

comparisons were performed between the different model types, and all differences were significant [Fs$(1, 9) > 10.2$, ps $< .02$] except the difference between the Dual-path model and the Linked-path model [$F(1, 9) = 4.2$; $p > .07$]. The large differences in the generalization abilities at epoch 4,000, when the training accuracy is the approximately the same, suggest that the architecture plays a crucial role in a model's ability to generalize.

Another point to notice is that the Dual-path model did not lose its generalization ability after it reached 99% accuracy on the training set. Instead, the model continued to improve, going from 76% at epoch 1,400 to 79% at the end of training (epoch 4,000). So, the Dual-path model seems to avoid overfitting the training set. Overfitting is a problem for generalization in error-based learning systems, especially when the model has too many weights (Hertz, Krogh, & Palmer, 1991). Normally, the better adapted a model is to the particular characteristics of the training data, the worse it becomes at dealing with new data. The Linked-path model may suffer from overfitting of the training set, because at epoch 1,400, its testing set accuracy reached asymptote, and began to decline. To test that the Linked-path model is overfitting, the difference in the sentence accuracy between epoch 1,600 and 4,000 was computed for all model subjects in each model type (epoch 1,600 was used to insure that all Linked-path models had reached asymptote). The mean difference was negative for the Linked-path model (-0.02), while it was positive for the other models (Prod-SRN $= 0.03$, No-event-semantics $= 0.004$, Dual-path $= 0.008$) (model type was significant, $F(3, 9) = 8.0$, $p < .0065$). Comparisons of these differences revealed that the Linked-path was worse than the Dual-path model [$F(1, 9) = 8.3$, $p < .02$]. Even though the Dual-path and the Linked-path models both maintained the same level of accuracy on the training set for this period, the Dual-path continued to improve and the Linked-path degenerated. A likely account of this is that the Linked-path model took advantage of the message information in the *what* units to help the model's sequencing system to memorize regularities in the training set. But this is the wrong thing to do if it wants to generalize to new sentences. The Dual-path model avoided overfitting because its isolation of lexical-semantics and sequencing kept message-specific knowledge from reducing generalization.

2.5. Dog-goal test

One part of symbolic generalization is the ability to bind words to novel event roles, and generate sentences that convey those novel meanings. In the training of the model, the grammar was constrained so that the word *dog* was never allowed to be the goal of the sentence. By testing the model on messages where the goal was bound to DOG, we could see whether the model can generalize its experience with other goals to produce these novel sentences correctly. One hundred test sentences were randomly generated with *dog* in the goal slot. Using the weights at epoch 4,000, all four model subjects for each of the four model architectures were tested on this dog-goal test set.

The dependent measure for this analysis was the percentage of sentences for which all the words match the target sentence exactly, or the overall sentence accuracy. The Prod-SRN model produced 6% of the dog-goal sentences correctly (Fig. 5). The other models generalized fairly well (No-event-semantics model 55%, Linked-path model 67%, Dual-path model 82%). An ANOVA was performed, and model type was significant [$F(3, 9) = 96.7$, $p < .0001$]. Pairwise comparisons showed that all differences were significant [Fs$(1, 9) > 6.6$, ps $< .03$].

The dog-goal test helps to explain why the Dual-path and Linked-path models were better than the other two models in the previous test of generalization using the 2,000 test sentences. While all the models achieved good accuracy at the sentences that they were trained on, it is likely that the novel test sentences had words in roles that had not been trained before (the training set was relatively small compared to the possible sentences that the grammar can generate). The Dual-path, Linked-path, and No-event-semantics models derived some benefit from the dual-pathways architecture, which allowed the same semantics (*what* units) to be used for different roles. So, if you learned to say *dog* in any of these models, there was a link from the semantic unit DOG to the word unit *dog*, and this allowed it to be said in different sentence positions. The Prod-SRN used a binding-by-space message representation, where different roles had their own set of semantic units. Since the training set did not include any sentences with dog as the goal, the semantics for *dog* in the goal slot (DOG3) had never been associated to any other units, so it would not be able to use this unit to activate the word *dog*, and this would keep it from producing it in the appropriate position. It also seems that

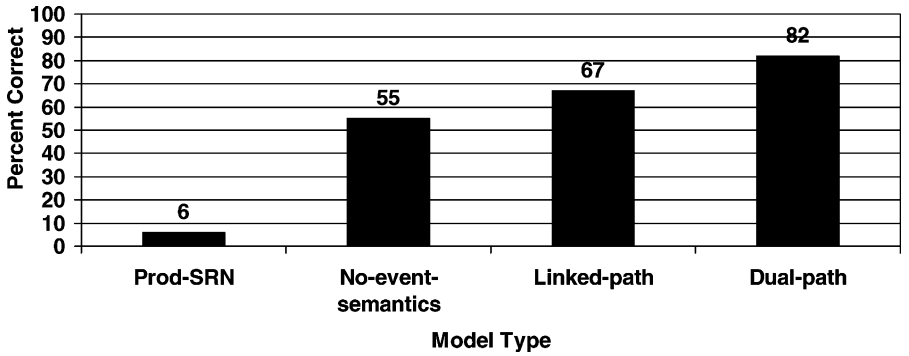

Fig. 5. Dog-goal test result.

the event-semantics information is crucial, because the Dual-path model was better than the No-event-semantics model. Event-semantics information helped the sequencing system know that the message had a goal and that the goal tended to occur in certain sentence positions. This helped the model sequence the GOAL *where* unit at the time when the goal should be produced. And, the difference between the Dual-path and Linked-path models suggests that the Linked-path model was including lexical-semantics in its dative syntactic representations (e.g., *dog* can't follow *to*), and that hurt its ability to produce novel dog-goal sentences. So, the dog-goal test showed how slot-independent lexical mapping, event-semantics information, and abstract syntactic frames work together to effect symbolic generalization.

2.6. Identity construction test

The dog-goal test was a good test of the ability of the models to generalize a word to a novel sentence position. But this test might overestimate the generalization abilities of the models, because the goal often occurred at the end of the sentence in many constructions, and so we do not know if the model would be able to continue to generate structure after producing a word in a novel position. Also, it could be that accidental distributional properties of the dative construction were influencing generalization. Consequently, another test was carried out. Inspired by Marcus's (1998) claim about the inability of SRNs to produce novel sentences like *a blicket is a blicket*, this identity construction will be used to see how well these models generalize. This identity test takes advantage of the accidental fact that in the random generation of each training set, only a subset of words used the identity construction. (Recall that existence and intransitive verbs were less frequent than other verbs in training.) Novel identity construction sentences were randomly generated for each model subject (the actual number varied between 48 and 58 because each training set had different identity sentences). The four model subjects for each model architecture were tested on these sentences at epoch 4,000 (Fig. 6). The difference between the Dual-path model and the other three models was quite dramatic. The Dual-path model had an 88% accuracy, while the other models do not get above 44%. An ANOVA was performed and model type was significant [$F(3, 9) = 5.1$, $p < .03$]. Pairwise comparisons found that the Dual-path model was superior to the Prod-SRN

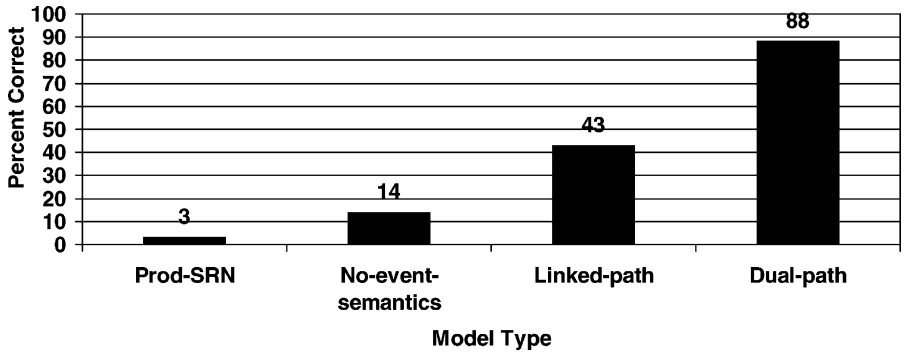

Fig. 6. Identity construction test results.

and No-event-semantics models [Fs$(1, 9) > 9.6$, ps $< .02$] and marginally superior to the Linked-path [$F(1, 9) = 3.6$, $p > .09$].

The fact that the Dual-path model does better on this test is not that surprising given the dog-goal test. What is surprising is how bad the other models are at producing novel words in this simple construction. In all the models, the identity construction used the patient slot to instantiate the single argument of this construction. All the models had experience mapping some words from the patient slot to both pre- and post-verbal sentence positions with other constructions. Even though they had this experience, they still were not able to make use of this information to help their generalization. Instead, they developed sequencing representations that were specific to the particular words that they had experienced in this construction. The Prod-SRN and the Linked-path models in particular were probably learning lexical-semantic-specific mappings, while the No-event-semantics model probably learned that the verb *is* was followed by the set of nouns experienced in training. The separation of lexical semantics from sequencing in the Dual-path model allowed the sequencing system to avoid using lexical-semantic information in its representation of the identity construction, and the event semantics helped the Dual-path model strongly activate the message–lexical system at the appropriate time to overwhelm any lexical-specific sequencing regularities that the sequencing system had picked up.

While the "identity construction test" is similar to Marcus's (1998) *A blicket is a . . .* test, it differs in that the models being compared here had previous experience placing the nouns into both surface positions. In Marcus's test, a novel word *blicket* is used; the human does not have experience placing the novel word in either sentence position. But even though the models have the previous experience placing nouns into these positions, the Prod-SRN and the No-event-semantics models cannot make use of this experience to increase their generalization. Of the models that I compared, only the Dual-path model gets this right.

2.7. Novel adjective-noun pairing test

Both of the two previous generalization tests involved placing a noun in a structure that it was not trained in. This ability is the natural outcome of incorporating variables that are bound to the semantics of phrases, because phrase ordering knowledge can generalize over these variables. One question is whether all symbolic generalization in the model requires that each element have its own variable. This question can be addressed by looking at novel sequences within noun phrases, because phrasal semantics was bound to a single *where* unit and therefore the phrase internal elements were not given their own variables. To test this in the model, I exploited restrictions on adjectives in the training grammar. The grammar restricted adjectives so that they could only pair with appropriate nouns. There were two kinds of adjectives, those that were restricted to animate entities (*nice, silly, funny, loud, quiet*) and those that were not restricted (*good, red, blue, pretty, young, old*). So, dogs could be nice, but cakes could not be nice. But both cakes and dogs could be old or good. In all the previous training and test sets, these restrictions were enforced. People, however, can make metaphorical extensions of animate adjectives to inanimate elements. For example, a car can be nice if it is easy to maintain. Or a wall can be silly if it is painted in a crazy fashion. We can see if the model can generalize symbolically without using separate variables if it can produce *nice car* or *silly wall* when the message calls for it.

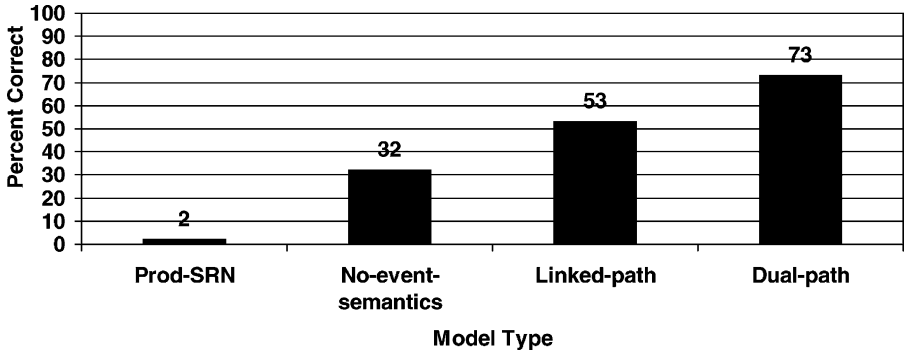

Fig. 7. Novel adjective–noun pairs test result.

One hundred randomly generated test sentences were generated with animate adjectives attached to inanimate nouns. All four model subjects for each model type were tested on this novel adjective-noun test set at epoch 4,000 (Fig. 7). Again, the Dual-path model was best, producing 73% of these sentences correctly. The Linked-path model produced 53%, No-event-semantics model produced only 32% correct, and the Prod-SRN was worst at 2% correct [$F(3, 9) = 33.1$, $p < .0001$, pairwise comparisons found that all differences were significant, Fs$(1, 9) > 6.9$, ps $< .03$]. The ability of the Dual-path model to generalize better than the other models in this case is not primarily due to the *what–where* system, as in the previous two tests. In the earlier tests, if the model produced the appropriate *where* unit at the right time, then the models would have a good chance of generalizing symbolically. Because both adjective and noun semantics (*what* units) were connected to the same *where* unit, the same strategy would not work in the case of phrase-internal sequences. Rather the model had to develop a way to sequence words in a symbolic manner without using variables.

For this to occur, the model had to get two things right. First, the appropriate *where* unit for the phrase had to be activated. The No-event-semantics model probably did not activate this *where* unit appropriately, because it did not know anything about the message. The second part was to sequence the words within a phrase without reference to their co-occurrence frequency. The Prod-SRN model should record lexical-specific co-occurrence frequency, because its hidden units have access to the semantics of the whole phrase, and so they will prefer that these adjectives be followed by animate nouns. The Dual-path and Linked-path models were able to meet both requirements for producing these novel phrases. Its *event-semantics* helped it activate the right *where* unit at the right time. And the compress units in the sequencing system kept the models from recording lexical-specific co-occurrence frequencies. The fact that these models can do this metaphorical extension suggests that the model has developed a symbolic ability to sequence words within phrases, in addition to its ability to symbolically sequence phrases within a sentence.

2.8. Conclusions about symbolic generalization in different architectures

Why was the Dual-path model better at generalization than the other three models? There were three dimensions that were manipulated in these comparisons. One was the message type.

The Prod-SRN had a slot-based message, while the other three models had the *what–where* representation. The *what–where* message allowed those models to learn syntactic structures which used the *where* units to activate variable information in the links. Over all the comparisons, the models with the *what–where* message were better than the Prod-SRN at generalizing, and so it seems that there is a definite benefit to using this type of variable representation.

The second dimension was the architecture of the network. The issue was whether a separation between the message and the syntactic representations was needed to achieve good generalization. This comparison can be seen in the differences between the Dual-path network and the Linked-path network. These networks were equivalent except that the Linked-path network linked the two pathways, and this allowed the syntactic representations to use the information about the message that was being produced. The Dual-path model was clearly better than the Linked-path model in the magnitude of all generalization measures (significantly better in two comparisons and marginally significant on another two). This comparison shows that the architecture plays a crucial role in keeping the model from learning the wrong representations for symbolic generalization.

The third dimension that was manipulated was the presence or absence of event-semantics information. While it is true that the Dual-path model did better than the No-event-semantics model on all measures, its higher performance cannot be solely due to the existence of event semantics. Recall that the Prod-SRN model's message representation also had this event-semantics information. In the Prod-SRN model, event semantics could help the learning of sequencing constraints, but it had to learn the individual combinatorial relationship between the event-semantics features in its message and each of the semantic-lexical mappings separately. In the Dual-path architecture, the event semantics was connected to an SRN that was blind to the message and able to sequence variables through the *where* units. So, in this network, the value of the event semantics was increased, because it was used to sequence variables within abstract frames.

Given that this paper addresses the limitations of connectionist models that Marcus (1998) points to, it is worthwhile to frame the model comparison in terms of his notion of a *training space*. Marcus argues that multi-layer connectionist models that use back-propagation do not generalize beyond their training space (Marcus, 2001). The training space is the set of input feature values that have been experienced during training. These input feature values have associated output outcomes, and so the model can interpolate between input values to find interpolated output values. But outside of the training space, these models cannot extrapolate to find appropriate output values. While all four model types received the same training set within a model subject, the architecture of the models created different training spaces for each model. The Prod-SRN model has a message where each role occupies different units. That means that its training space is role-dependent, where each word's semantics has to be trained in a particular role to generalize appropriately. In the other three models, there is only one set of *what* units that represents lexical semantics for all the event roles. If the model learns to produce a word correctly, then that word's semantics is in the training space. And because the Linked-path and Dual-path models have the event semantic units, the ability to produce one sentence in a construction correctly with the event semantic inputs allows other sentences in that construction to be in the training space. The problem with the Linked-path model is that the sequence representations that it uses are contaminated with lexical-semantic information, because of the

link between the message–lexical system and the sequencing system. The Dual-path model overcomes this limitation by isolating these two systems, forcing the sequencing system to only use a limited number of syntactic categories to make the distinctions that will be useful. The Dual-path model is successful at capturing the character of human sentence generalization because the training space of the Dual-path model is divided into constructions (which operate on syntactic categories and variables) and lexical-semantic representations (which select words), and this way of dividing up language seems to be appropriate for characterizing human language use.

The most surprising aspect of this model comparison is the relationship between variables and symbolic generalization. From the literature on symbolic generalization (Fodor & Pylyshyn, 1988), you might expect a straightforward relationship between the existence of variables and the ability to generalize to novel elements. But the novel adjective–noun pairings show that symbolic generalization can arise without separate variables for each element, and the No-event-semantics model showed that variables without meaning do not generalize well. These comparisons suggest that symbolic generalization in language production is really several separate types of generalization. Interaction between variables and syntactic categories will yield one type of novel pairings, while event semantics and variables will yield a different kind.

3. Hidden units analysis

In order to understand how each pathway in the Dual-path model works, it is valuable to examine the activation of the units in the model as they process sentences. It is useful to look at the *compress* units to understand the sequencing system, because these units directly influence the production of words, and so the effects of the sequencing system on words must be propagated through this layer. To see how the message–lexical system works, it is not as useful to look at corresponding input to the *word* units, the *what* units, because the *what* units depend on the message-specific *where–what* links. So, I looked instead at the activation of the *where* units, which were less message-dependent. As a single Dual-path model was tested on the novel 2,000 sentences test set at end of training (epoch 4,000), the activation of the *where* and the output *compress* units was recorded. There were four *where* units (agent, patient, goal, action) and 10 *compress* units. The average activation of these units for one model subject when tested on the 2,000 novel sentences was calculated, and the results were quantized into five distinct levels, to make the similarities between units more evident (Table 5). Because of this averaging, the most diagnostic information comes from the strongly activated units (dark elements in table), because the less activated units could reflect the averaging of strong and weak elements over different sentences. The activation of these units was averaged by syntactic class with verbs separated by verb class (intransitive, transitive, psych, change-of-state, cause–motion, spray–load, dative).

First consider the *compress* units, which represented the output of the sequencing system. Here, the goal was to test the claim that these represent syntax-like states in the model. While the activation was quite distributed, there were some clear patterns. Verbs mainly used the units C1, C2, C4, C5, C10 to activate the appropriate verb, and similar verb classes

had similar activation patterns. C6 and C8 seemed to be more specialized for phrasal elements like nouns, adjectives, prepositions, and determiners. Nouns and adjectives shared the same units except nouns also activated C4. Determiners and prepositions both had many units activated. Auxillary and intransitive verbs shared C3 and C5. Syntactic categories that had more activated units in this layer depended on the sequential system more than other categories, and therefore closed class words seemed to depend on this pathway more than open class elements. Thus, the *compress* units use distributed representations to encoded important syntactic knowledge such as major syntactic category distinctions and verb class information.

The activations of the *where* units as a function of syntactic category tell a different story (Table 5, right side). The *where* representations were not strongly differentiating these categories. The action role (AC) was active for verbs, prepositions, determiners, and nouns. The agent role (AG) was active for determiners, nouns, adjectives, auxillaries, and intransitive verbs, and the patient role (PA) had a similar pattern except prepositions were activated by this role. The goal role (GL) was active for prepositions only. So, some syntactic information is available, but the distinctions among the categories is not strong. Also, verb class

Table 5
Averaged activations of *compress* units and *where* units by syntactic category[a]

Syntactic categories and verb classes	*Compress* units										*Where* units			
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	AC	AG	PA	GL
Determiners	■	■	■	■	■	■		■	■	■	■	■	■	
Adjectives	■					■		■				■	■	
Nouns	■			■		■		■			■	■	■	
Prepositions		■	■	■		■	■	■		■	■		■	■
Auxillaries			■		■						■	■	■	
Intransitive verbs			■	■	■	■				■	■	■	■	
Transitive verbs	■			■	■					■	■			
Psych verbs	■			■	■					■	■			
Change-of-state verbs	■	■		■	■			■		■	■			
Cause-motion verbs	■	■									■			
Spray-load alternation verbs	■	■		■	■						■			
Dative alternation verbs	■	■		■					■	■	■		■	

[a] Five levels of activation: black = high, white = low.

distinctions were not maintained in these units. So, it would seem that the sequencing system, rather than the *what–where* system, was responsible for much of the syntactic behavior of the model.

To understand how the *where* units influence processing, a second analysis was done, looking at activation of these units given a particular sequence of syntactic categories. In English, sequences of syntactic categories encode role information, and so the *where* units should be more defined when grouped on the basis of the preceding sequence. The unit activations came from a single Dual-path model tested on the 2,000 novel sentences at epoch 4,000. In Table 6, the average activation of units is given for a syntactic category in a particular sequence (marked in bold). For example, if we had a sentence *The man gave a cake to the cat*, the state of the *compress* and *where* units would be recorded for each word in the sentence. These states were averaged over the sentences with similar sequences of syntactic categories in the novel 2,000 sentences test set to get an average *DET NOUN VDAT DET NOUN PREP DET NOUN* state representation, and this was placed into Table 6. At the end of the table, the average activation for a prepositional dative sentence with the word *dog* in the prepositional phrase was appended for comparison with other nouns. Two other lines were also appended to show the average activation for prepositional phrases with an adjective.

Table 6
Average activation of *compress* and *where* units predicated on syntactic sequences

Syntactic categories sequence (**bold** marks position)	*Compress* units										*Where* units			
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	AC	AG	PA	GL
DET N VDAT DET N PREP DET N	0.5	1.0	1.0	1.0	1.0	0.0	0.0	1.0	1.0	1.0	0.5	1.0	0.0	0.0
DET **N** VDAT DET N PREP DET N	1.0	0.2	0.0	0.2	0.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0
DET N **VDAT** DET N PREP DET N	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.5	0.0	1.0	0.0	0.0	0.0
DET N VDAT **DET** N PREP DET N	0.5	1.0	1.0	0.2	0.5	0.2	0.0	1.0	1.0	1.0	1.0	0.0	1.0	0.2
DET N VDAT DET **N** PREP DET N	0.5	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.2	0.0	1.0	0.0
DET N VDAT DET N **PREP** DET N	0.0	0.5	1.0	1.0	0.0	0.0	1.0	1.0	0.5	0.5	1.0	0.0	1.0	0.5
DET N VDAT DET N PREP **DET** N	0.5	1.0	1.0	1.0	0.5	1.0	0.0	1.0	1.0	1.0	0.2	0.0	0.0	1.0
DET N VDAT DET N PREP DET **N**	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0
DET N VDAT DET N PREP DET **DOG**	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0
DET N VDAT DET N PREP DET **ADJ** N	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0
DET N VDAT DET N PREP DET ADJ **N**	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	0.5	0.0	0.0	1.0

As the model produced the sentences, the activation of the *where* units tracked the phrases that the model was producing (the sentence *The man give the cake to the cat* will be used as an example). As it produced the subject *DET N* (e.g., the man), the agent (AG) unit was activated strongly. Then it turned off as the action unit (AC) was activated and the dative verb (VDAT) (e.g., *give*) was produced. The next noun phrase started with both patient (PA) and goal (GL) activated, but as the patient phrase won (e.g., *the cake*), the GL unit was shut off. This demonstrated the incremental nature of the model's decisions about structure selection, because if it had planned the sentence structure earlier, GL would be deactivated from the beginning of the production of the patient phrase. Once the patient phrase and the preposition were produced, the model produced the goal phrase by activating the GL unit (e.g., *the cat*). This GL unit stayed activated through the whole phrase. Because the lexical-semantic information was embedded in *where–what* links, the sequential activation of *where* units is exactly the behavior that we would expect in order to extract this information at the appropriate moment.

There is some independent support for thinking of sentence production as a process that involves moving attention over event roles, or the sequential activation and deactivation of roles from the study by Griffin and Bock (2000) mentioned earlier. They found that when speakers describe pictures of events, they tend to fixate on the picture elements right before naming them in their sentences. They found that this fixation depended on the syntactic structure that they actually used, which suggested that the syntactic structure and eye movements are linked in some manner. Production theories with static message representations (e.g., Chang et al., 2000) would not predict that eye movements would be so synchronized with structural decisions. The Dual-path model, however, used message representations that were spatially represented in *where–what* links, and which were dynamically activated during production. During event description, activation of the appropriate *where* unit might be related to focusing attention on elements in a scene. If this were the case, then both structural decisions and eye movements would be related to the activation of *where* units, and this provides a reason why syntax and spatial processing might be synchronized.

Incrementality can also be seen within noun phrases. When the model produces noun phrases that have an adjective (DET ADJ N) versus those that don't (DET N), the activation values for the word after the determiner (DET) do not seem to differ. In bottom of Table 6, the activation states for both the *compress* units and *where* units were identical for the production of the noun in the sequence . . . *PREP DET* ***N*** (e.g., *cat*) and the production of the adjective in the sequence . . . *PREP DET* ***ADJ*** *N* (e.g., *red*). What this shows is that the model did not plan to produce either the adjective or noun specifically at the point after the determiner. Rather, it simply produces the word that was most activated at that point in the sentence. If the adjective was produced, then the model activated C4 and deactivated C1 to produce the noun at the next timestep. If the noun was produced, then the model was done producing the sentence. So, the model was being incremental at various points in processing, which is consistent with experimental work in language production demonstrating that sentence construction is sensitive to the lexical availability of words at different points in processing (Bock, 1986; Ferreira & Dell, 2000).

The hidden unit analysis can also help us understand how the model generalized words to novel positions as in the dog-goal test. When we look at the activation pattern for nouns that have been trained in the goal role (. . . *PREP DET N*), it is no different for nouns that were never

trained in that role (. . . *PREP DET DOG*). The model learned to treat this novel message in a way that was identical to the other messages in this construction. Because of the architecture of the Dual-path network, this ability was due to the *event-semantics* units, in concert with the *cwhere* information, activating the goal unit at the appropriate moment in the sentence. The mapping from *event-semantics* units to *where* units was not novel (it was shared with all dative sentences), so the model could sequence any word that was attached to the goal unit. The equivalent mapping in the Prod-SRN involved mapping from role-specific semantic units like DOG3 in the goal slot to the appropriate sentence position. DOG1 and DOG2 had been trained before, but DOG3 had never been used before, and that was why the Prod-SRN model failed to generalize properly. DOG3 was not in the training space of the Prod-SRN model, because it was not explicitly trained. So, the Dual-path model uses the event semantic information to select the appropriate sequence of *where* unit activation for both sentences that it has produced before, as well as sentences with novel lexical items.

The hidden unit analysis tells us several things about the Dual-path model. First is that syntactic categories are represented primarily in the sequence system using distributed representations, while the activation of the *where* system seems to reflect the target phrase that is being produced. Second, processing in the model is incremental, and this incrementality can be seen in the way that lexical factors influence structure selection, and the sequencing of the *where* roles. The third point is that the model seems to be treating novel sentences in a way that is identical to the way that other sentences in that construction are treated.

4. Constraining overgeneralization: Baker's paradox

In the previous sections, I concentrated on the computational properties of the model and how these computational properties led to symbolic behavior. Humans exhibit symbolic behaviors, but these behaviors also seem to be constrained by statistical regularities. Since the Dual-path model implements symbolic processing with a statistical learning mechanism, we should be able to see the influence of these types of regularities on some aspect of the model's behavior in a way that is functionally similar to some aspect of human behavior.

A useful domain to look at the role of statistical processing is the way that verbs are paired with structural frames. Unlike nouns, verbs seem to be more selective about the structures that they can be paired with, and this relationship seems to be probabilistic, that is, it is graded. While nouns are easily paired with sentence frames, verbs are less easily paired with frames that they have not been heard in (Tomasello, Akhtar, Dodson, & Renau, 1997). The problem of constraining verb generalization is a problem for symbolic systems, because verbs and nouns are both controlled by variables. The same mechanism that gives nouns their ability to generalize to different frames might, one would think, also give verbs the same abilities. This property of symbolic systems has led to a learnability problem that was first described by Baker (1979), and which is referred to as Baker's paradox (Gropen et al., 1989). The paradox arises from the fact that children both seem to be able to overgeneralize a verb to a novel frame and yet they are reluctant to do so. This behavior could be explained if children started with a tendency to overgeneralize, and gradually learned to constrain that generalization because of negative evidence from their parents. But adults do not give enough detailed direct negative

evidence for children to avoid overgeneralization, and so it is a puzzle how they learn to constrain themselves.

The Dual-path model implemented symbolic processing in a framework that used a statistical learning algorithm, and so the same questions about verb generalization could be applied to the model. If the Dual-path model is simply a symbolic system that generalizes freely, then it may also be subject to Baker's paradox, because the training set does not provide the type of information necessary to restrict generalization. Specifically, the model never received negative evidence that verbs could not occur in alternative constructions, so we would expect that all verbs would generalize equally well. If, on the other hand, the model is simply a statistical learning system, then we might expect that verbs would not generalize to novel frames, because these novel pairings have a statistical frequency of zero. But if the Dual-path model employs the right mix of these symbolic and statistical properties, it should exhibit properly constrained generalization.

The experimental evidence for Baker's paradox can be seen in Gropen et al.'s (1989) experiments. In their third experiment, they taught a novel verb in a neutral frame while demonstrating a transfer action (e.g., X moves Y to Z while saying *This is norping*). They then tested the child's ability to produce the novel verb in a double object frame (e.g., *You norp me the ball*). They also asked the child to describe the action using a known dative verb (e.g., *You give me the ball*). They elicited 78% double object responses for verbs that the child knew before the experiment (e.g., *give*) and 41% double object responses for novel verbs (e.g., *norp*). Since *norp* has never been associated with the double object frame, the child has no statistical evidence that *norp* can go in that frame, and they should not be able to produce any double object responses if this evidence is the sole basis for use. But that is incorrect. If the way that children generalize is to use a variable (ACTION = *norp*) and a frame with variables (*AGENT ACTION GOAL PATIENT*), then we would predict that they would use *norp* as much as other verbs that they can use in that structure (e.g., *give*). But they don't. The fact that *norp* generalizes at an intermediate level suggests that neither of these accounts, by itself, can explain the generalization.

To examine Baker's paradox in the model, 30 messages that would produce double object dative structures using the verb *throw* (e.g., *The boy throw the girl a cup*) were generated. Several other lists were created by replacing the action semantics of the *throw* sentences with the verbs *dance*, *hit*, *chase*, *surprise*, *pour*, and *load*. In training, only *throw* occurred in the double object frame. The other verbs never occurred in this frame. *Dance* only occurred in the intransitive construction. *Hit*, *chase*, *surprise* occurred only in transitive and benefactive frames. The verb *pour* occurred only in the cause–motion frame. The verb *load* occurred only in the cause–motion frame and the change-of-state frame. To create a double object dative test set, the goal argument was made more prominent (by setting the event semantics unit TRANSFER to be more active than MOTION), which made the double object the target structure. All four model subjects for each model type were tested with all seven of these test sets.

Fig. 8 shows the average sentence accuracy on the seven test sets in the Dual-path model at epochs 1,000, 2,000, 3,000 and 4,000. For the verb *throw*, which was trained with the double object structure, the model achieved a high level of accuracy (above 95%) after 2,000 epochs. The other verbs were not trained in this structure. *Pour* generalized well to this structure above 78% after 2,000 epochs. *Load* generalized at 86% at 2,000 epochs, but fell to 56% by 4,000 epochs. The verbs *surprise*, *chase*, and *hit* generalized above 27% at 1,000 epochs, but fell to

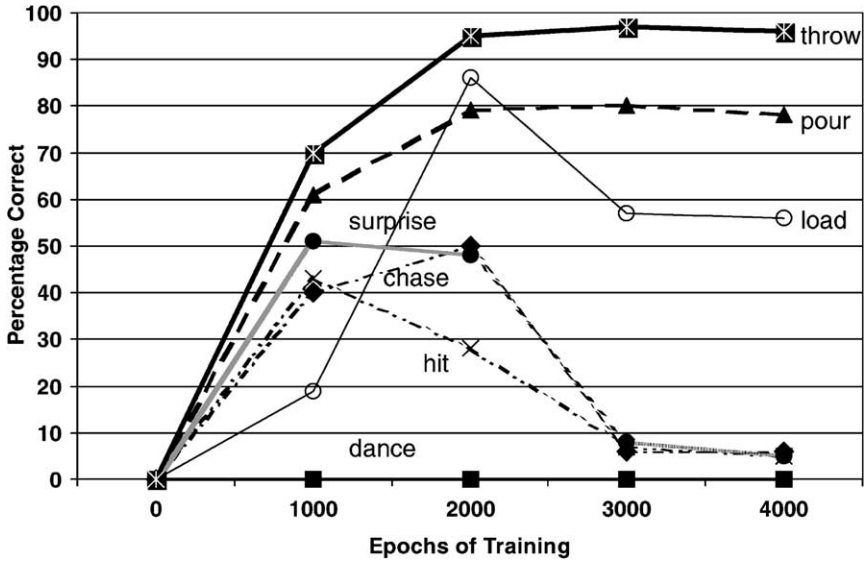

Fig. 8. Accuracy of generalization to double object dative frames.

5% at 4,000 epochs. The verb *dance* never generalized to the double object structure. Although none of the test verbs except *throw* had training in the double object structure, there were several different degrees of overgeneralization. Some verbs generalized in a free variable-like manner (e.g., *pour*), while others generalized in a way that reflected the co-occurrence properties of the construction and that verb (e.g., *dance*). If we compare the model with the children, we can look at epoch 1,000, which approximates the model's childhood state. At epoch 1,000, the model produced 70% double objects for verbs that it had experienced in the double object dative frame (e.g., *throw*) and 36% double objects for verbs that had never appeared before in this structure (average of *chase*, *dance*, *hit*, *load*, *pour*, *surprise*), which shows that the model can capture the intermediate nature of generalization that Gropen et al. found.

The developmental pattern of the model also resembled the way that generalization changes in children. The model initially was unable to produce any sentences, but as it learned the language, it started to overgeneralize between epoch 1,000 and epoch 2,000. This overgeneralization was slowly reduced as the model continued to learn. *Surprise*, *chase*, *hit*, and *load* showed this pattern. The pattern was partially due to differences in the speed that the two pathways in the model learned their corresponding representations. The mapping from event semantics through the message–lexical system is shared with all the sentences in a construction, and so it is learned quickly. This allowed the model to overgeneralize. The mapping from the *cword* units through the sequencing system to the *word* units requires lexical-specific learning, and this takes longer to constrain generalization. This knowledge eventually reduced the overgeneralization of the model. These mappings resemble the *broad* and *narrow* constraints that Pinker (1989) has argued for. The mapping of the *event-semantics* units to the *where* units is similar to the operation of the broad constraints, where the semantics of the whole construction influences the order of the arguments (Gropen et al., 1989). The mapping from the *cword*

units to the *word* units through the sequencing system represents the operation of the narrow constraints, which involves the way that lexically-specific classes restrict the generalization of the construction (Brooks & Tomasello, 1999).

The reason for the variability in the model among different verbs (e.g., *dance* vs. *pour*) in the degree of their ability to generalize to the double object dative structure was partially due to overlap in event semantics. *Dance* and *throw* shared No-event-semantics, and so it was very difficult to produce *dance* in a dative frame. *Pour* generalized well to the double object frame, because it shared the features CAUSE and MOTION with the dative construction. The link between event semantics and the verb is probably one of the main reasons that the children in Gropen et al. (1989) generalized, because *this is norping* was presented in a context which should have suggested the event semantics for a transfer event, which was also present at test. Another reason for variability stems from the available syntactic frames that a verb was seen in. For example, *pour* and *load* shared the same features CAUSE and MOTION with *throw*. But *load* can also occur in the change-of-state construction (e.g., *The boy loaded the wagon with hay*) where the goal (entity that undergoes a state-change) occurs after the verb. Initially, *load* overgeneralized to the dative as much as *pour* does (around 82%). But after epoch 2,000, as the model learned to use the change-of-state construction to put the goal after the verb, its ability to use the double object dative was reduced (at epoch 4,000, *pour* 78%, *load* 56%). Since the change-of-state construction puts the goal after the verb, it is more similar to the double object datives which also puts the goal after the verb. Because of this similarity, the ability to be used with the change-of-state construction should interfere with the ability of this verb to be used in double object dative. The change-of-state construction is said to preempt the use of the double object construction as a way of fronting the goal. Preemption or blocking is an important way that children reduce overgeneralization (Clark, 1987; Pinker, 1989). Another important reason for variability in the model's generalization was due to the simplicity of the model's verb representations. Lexical semantic similarity was not captured in the model (e.g., *eat* and *drink* shared no semantic features in their *what* unit representations), so event semantics (e.g., TRANSFER) provided the only reliable information about generalization. In people, verbs cluster into semantic classes that are smaller than the broad classes specified by event semantics, and these subclasses are predictive of syntactic frames that they can appear in (Fisher et al., 1991).

While the model does seem to be subject to Baker's paradox, it constrains generalization in a way that is similar in character to the operations that children might be using. It overgeneralizes by making use of the pathway through the message–lexical system, and it constrains generalization by learning lexical-specific information in the sequencing pathway. While the results are promising, it is likely that children are exposed to a larger collection of words and structures, and future work will need to address how well these types of models scale up to the language that real children experience.

5. Dissociating processing systems in aphasia

Connectionist models have been used to link our understanding of normal language processing to cases where brain damage has impaired critical processing systems (Dell, Schwartz,

Martin, Saffran, & Gagnon, 1997; Plaut, McClelland, Seidenberg, & Patterson, 1996) and these studies have helped us understand how the architecture of a language processing system can influence the type of symptoms that appear in impaired patients. In describing the architecture of the Dual-path model, I have concentrated on the way the architecture enables the model to exhibit certain functional behaviors. But it is also desirable to show that this model approximates the architecture of language in the brain. This can be done by establishing that damage to the physical architecture of the model can lead to symptoms that are similar to patients with injury to real brain systems. To do this, lesions will be applied to the two separate pathways, and the resulting behavioral effects will be analyzed. These behavioral effects will be compared with aphasic symptoms, to see if the model's processing abilities are damaged in ways that are similar to patients with brain injuries.

Double dissociations in the production of different lexical categories have been an important type of evidence for separate processing systems. Researchers have suggested that function words (prepositions, determiners, auxillary verbs) and content words (nouns, adjectives, verbs) are represented in separate systems (Goodglass & Kaplan, 1983). Some patients have more difficulty with function words and relatively less difficulty with content words. Other patients have the opposite pattern, with content words being relatively spared and function words being relatively impaired. Other researchers have found that light and heavy verbs also dissociate (Breedin, Saffran, & Schwartz, 1998). Light verbs (such as *go*, *give*, *have*, *do*, *get*, *make*, and *take*) are the first to be learned, are the most frequent in the speech of children, and across languages are the first learned (Clark, 1978). Some aphasic patients have trouble with heavy verbs, and are relatively spared with respect to light verbs (Berndt, Mitchum, Haendiges, & Sandson, 1997). Other patients have the reverse pattern (Breedin et al., 1998). These double dissociations are important, because they demonstrate that each behavioral pattern is dependent on different brain areas, in that there exists a way to focally lesion each system without automatically impairing the other.

Gordon and Dell (2002) argue that the function-content word dissociation and the light-heavy verb dissociation reflect an underlying distinction between syntactic and semantic representations, and lexical items are dependent on these separate representations to different degrees. Using a two-layer connectionist model that learned to produce simple sentences, they showed that the model learned representations in which light verbs relied more on the syntactic system and heavy verbs depended more on the semantic system. Since light and heavy verbs both had syntactic and semantic determinants, these dissociations in the model arose out of differences in the degree of dependence on each system. Because the Dual-path model also claims that different types of representations (as a result of different pathways) are independently influencing lexical representation, it is possible to examine whether this model will also exhibit aphasic dissociations.

To test the importance of separate pathways in the trained Dual-path model, lesions were applied to each of the pathways to create two types of impaired models. The *what-word* lesioned model (WWL model) was created by lesioning the message–lexical system, specifically the links between the *what* units and the *word* units. The *hidden-word* lesioned model (HWL model) was created by damaging the sequencing system by lesioning the links between the *hidden* units and the *word* units. Lesions were created by randomly removing weights between sets of units. Each of the four trained Dual-path model subjects received four lesions: two

Table 7
Sample output of what-word lesioned model (WWL) and hidden-word lesioned model (HWL)[a]

(1)	Target	a man eat the icecream
	WWL	a man make the blue
	HWL	eat man the icecream
(2)	Target	a cup scare a woman
	WWL	a – scare a –
	HWL	cup scare woman
(3)	Target	the man pour a silly dog in a boy
	WWL	the blue man put a cup
	HWL	the man pour silly silly silly silly boy
(4)	Target	a girl give a woman the croissant
	WWL	a grass give a – the –
	HWL	girl a girl put croissant the the croissant
(5)	Target	the owl throw the boy the cafe
	WWL	the blue woman give the blue – the blue
	HWL	the owl throw the the the the the the

[a] Dashes (–) mark positions where all output word units were less than the threshold of 0.5.

for the *what–where* links and two for the *hidden-word* links. These 16 models were tested on the 2,000 novel sentences test set and the results were coded for analysis. Lesioning of the *what-word* links was more damaging to the network than the same amount of lesioning of the *hidden-word* links. To reduce the differences due to overall severity of the lesion, the *hidden-word* lesion removed 11% of its connections, while the *what-word* lesion only affected 5% of its links. These lesions led to the same average word accuracy (correct word in the correct position) over the model subjects for both WWL and HWL models (45.9 and 46.1%, respectively, not significantly different, $F(1, 7) = .008$, $p > .9$), which suggests that the overall severity of each lesion was equal. Word accuracy is a finer measure of model accuracy that can capture partial productivity that is lost with grosser measures like sentence accuracy. Table 7 shows the intended target and the output of the two lesioned models when they try to produce this target. This sample illustrates some of the differences in the way the lesions influence production. For example, the WWL model will get local word ordering correct, but will often omit content words and substitute non-contextual words (e.g., *girl* becomes *grass* in 4). The HWL model tends to convey the appropriate content, but sometimes in the wrong order (1), without determiners (2), and repeating content words (4). Determiner use by the WWL model is more frequent in the appropriate positions, and sometimes that model repeats multiword sequences (in 5, it says *the blue* twice), while the HWL model only repeats single words. The WWL model replaces heavy verbs with their light counterparts (*pour* → *put*, *throw* → *give*). Broadly speaking, the HWL model is acting like a Broca aphasic, and the WWL model is acting like a Wernicke aphasic, although in humans and in the models, there is quite a bit of variability.

To examine the use of function and content words, the percentage of function words that were correctly produced and the corresponding percentage of content words will be the dependent

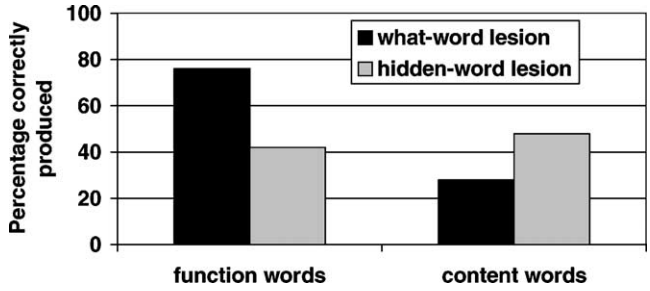

Fig. 9. Accuracy at producing function and content words depending on lesion type.

measures. Function words in the model include prepositions, determiners, and the auxillary verbs. Content words constituted all other words. As shown in Fig. 9, the WWL model produced function words better than the HWL model (76% and 42%, respectively) [$F(1, 7) = 8.6$, $p < .02$]. The reverse was true for the content words, with the WWL producing fewer correct words (28%) than the HWL model (48%) [$F(1, 14) = 12.1$, $p < .02$]. This double dissociation was a natural outcome of the constraints of the Dual-path architecture. Content words, by definition, have content, or meaning, and so they depend more on the message and the *what-word* pathway. Function words were only produced in certain syntactic contexts, and so they needed the syntactic information that is provided by the sequential system. Lesioning each of these pathways selectively damaged one component, leaving the other relatively spared.

The other dissociation that was examined in the model was the light–heavy verb dissociation. Several theories of verb semantics have argued that the light verbs represent basic primitives of sentence meaning (Goldberg, 1995). In the model, I have incorporated these ideas by treating some verbs as the default verb for a construction (these verbs are marked as bold in Table 2). This means that these verbs do not have features in the action event role. For example, the verb *throw* had a feature in the action event role, but the verb *give* did not. Because of this difference in the features in the *what–where* links, the model should depend more on the message–lexical system for heavy verbs, and more on the sequencing system for the light verbs. As shown in Fig. 10, the WWL model produced light verbs correctly 79% of the time, while the HWL model produced them only 29% of the time [$F(1, 7) = 33.3$, $p < .0007$]. For heavy verbs, the WWL model (15%) was more impaired than the HWL model (53%) [$F(1, 7) = 44.4$,

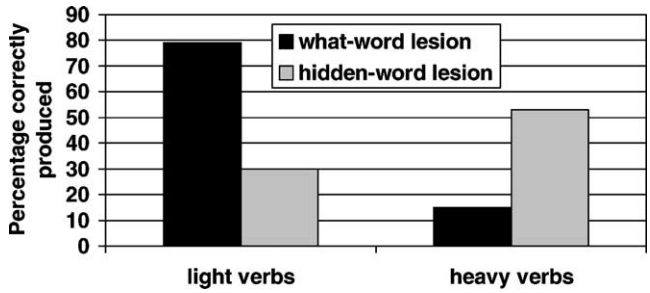

Fig. 10. Accuracy at producing verbs depending on lesion type.

$p < .0003$]. So, the model exhibited a double dissociation for verb complexity, as has been found in the aphasic literature.

For function/content word use and light/heavy verb use, the model exhibited double dissociations that have been argued to reflect selective impairment of processing modules in aphasic brains. In the model, these modules were given concrete instantiations and have been shown to work together to produce sentences. One module corresponded to the message–lexical system (impaired in the WWL model), which supported the production of semantically rich information like content words and heavy verbs. The other module was the sequencing system (impaired in the HWL model), which supports categories that were identified with syntactic frames like function words and light verbs. The original motivation for these two modules was the computational demands of getting a connectionist model to generalize more symbolically. But the solution to that problem also nicely accounts for these aphasic dissociations.

In addition to its ability to model some dissociations in aphasia, the Dual-path model is broadly consistent with some theories of how language knowledge is distributed in the brain, which is not surprising since most theories of brain localization of language are designed to account for aphasic symptoms. Before describing these theories, it is important to clarify the relationship between computational models and data about brain connectivity and localization of function. Since brains are made up of immensely complex networks that are dynamic, it is unlikely that a computational model with a few hundred neurons can provide a satisfactory model of a whole neural system. But, if one is comparing the properties of different computational models, evidence about brain connectivity and localization of function can provide converging evidence supporting a particular architecture.

Here, I will use evidence from neuropsychological studies in a comparison of the Prod-SRN and the Dual-path model. One difference between these two architectures is that the Dual-path model's sequencing representations are isolated from the message, while the Prod-SRN model's sequencing representations are linked to the message. One theory of language localization that supports the Dual-path model is one that Ullman (2001) has proposed. He argues from a variety of behavioral and neuropsychological data that the way that language is represented in the frontal and temporal lobes differs. The left temporal lobe tends to encode semantic/episodic knowledge and the left frontal lobe is more likely to encode abstract language rules. The Prod-SRN model does not support the separation of these types of information into separate lobes, because its semantic knowledge in the message can influence its sequencing representations directly. The Dual-path model naturally models this separation, where the message–lexical system acts like the temporal lobe and the sequencing system acts like the frontal lobe. Furthermore, the original evidence for the what–where distinction, localization of separate pathways for object and spatial processing (Milner & Goodale, 1995; Mishkin & Ungerleider, 1982), and evidence that they are bound in the temporal lobe (Karnath, 2001) is more consistent with the approach of the Dual-path model which binds object and spatial roles in the message–lexical system, rather than the Prod-SRN model, which does not separate objects from their roles.

Whether the architecture of Dual-path model appropriately captures the functional relationships between brain architecture and language is an issue for further research. But it suggests that incorporating some distinctions that have been found in the brain can be helpful in both modeling the effects of brain injury and in explaining how the architecture of the brain contributes to the symbolic creativity of language.

6. Conclusions

In summary, an incremental connectionist model was able to learn to produce sentences from message-sentence pairs, and to generalize that knowledge to novel sentence structures. It accomplished this generalization by isolating different types of knowledge in each of its two pathways, and then creating novel combinations from the pathways. Sometimes this involved putting a word into a novel position (dog-goal, identity, and novel adjective-noun pairing tests), and sometimes it involved limiting generalization through sensitivity to the statistical regularities of lexical items (novel verb–frame pairings). Furthermore, the architecture of the model exhibited double dissociations that resemble dissociations that are found in aphasia.

A key innovation in this work was the use of weights to represent temporary variable bindings in messages, and the use of a sequencing network to activate the variables. Instead of arguing that the variable-like behavior of language emerges from distributed representations and statistical learning, this model instantiates the idea that variable-like behavior arises from pre-existing variables in the spatial system that are linked to sequencing representations in the language system. Given that space and language are proposed to be related by behavioral scientists (linguists, psycholinguists, developmental psychologists) and are proposed to inhabit similar regions in the temporal lobe, it seems natural that combining spatial variables with connectionist learning algorithms would yield a system that treats language in a manner that is more similar to human speakers.

Do these symbolic abilities make the model too powerful? The model has parameters that allow it to vary its dependence on symbolic and statistical processing. For example, the size of the compress layer determines how much the sequence system can influence lexical selection, and so the more compress units the model has, the more statistical sequencing information influences production. But, being able to process sentences that do not conform to the experiences of the model is an important feature of the model. Chomsky's (1957) well-known claim that English speakers can recognize the grammaticality of meaningless sentences like *colorless green ideas sleep furiously* is evidence that we do not simply construct sentences from only statistical or semantic regularities. The model's ability to produce novel adjective–noun pairings (as in *green ideas*) and violations of the lexical experience of verbs (as in *ideas sleep*) is a reflection of its ability to use finite means to generate a greater set of possibilities.[1]

Part of the power of the *what–where* variable representations came from the fact that they are linked to abstract structural frames. These frames arose from learning to combine different types of information to predict sequences, and hence the abstractness of these structural frames depended on keeping message–lexical content separate from the sequencing system, as in the Dual-path model. The abstractness of these frames does help symbolic generalization in this model, allowing any member of a syntactic category to operate in a certain position, if the message is set appropriately. But, at the same time, these frames are not much use if there is little concrete information that can be used to decide when to use these frames. In the model, the event semantics do this job. So, while the architecture increases the abstractness of the syntactic representations, the event semantics give these structures specific conditions for their use.

The combination of variables, syntactic structural frames, and event semantics is powerful. This combination puts this work somewhat outside the two main schools of connectionism. One school, the eliminative connectionist approach, emphasizes the power of error-based learning

algorithms to extract statistical regularities from the training environment (Elman, 1993; Plunkett & Juola, 1999; Rohde & Plaut, 1999). At the same time, this school de-emphasizes the role of the input representations and the architecture of the model. While the Dual-path model does make use of statistical regularities that are extracted by the learning algorithm (particularly in the sequencing system), a large emphasis is placed on pre-existing representations (*what–where* representations) and the architecture of the network (Dual-path architecture) in explaining how the model works. The other school, the structural connectionist approach, takes a pragmatic approach to building networks, focusing on particular computational operations like variable binding (Shastri & Ajjanagadde, 1993) or task decomposition (Jacobs, Jordan, & Barto, 1991). The Dual-path model shares the pragmatic approach to building networks and the emphasis on using modules to build more complex systems, but it also places a large emphasis on the way that different systems interact with each other, and the way that learning is important for both learning system-internal representations (e.g., sequence system weights are learned), as well as how systems interact (e.g., combining the outputs of the message–lexical and sequence systems, which is crucial for the light/heavy verb dissociations). The Dual-path model is a compromise between these two approaches, incorporating aspects of both into a single framework.

In addition to suggesting how eliminative and structural connectionism could be combined, this modeling work illustrated the usefulness of building models that attempt to link different domains. Psycholinguistic research suggested that we needed connectionist-learning algorithms and architectures to capture detailed lexical statistical regularities that arise in incremental processing (MacDonald et al., 1994; MacWhinney, 1987). Insights from biological, developmental, and linguistic theories of spatial representations hinted at how messages could be structured, which led to the *what–where* message. This message representation, in turn, necessitated the dual-system architecture in order to place symbolic processing within a connectionist framework. The Dual-system architecture allowed the model to constrain verb generalization (Baker's paradox) and to account for certain double dissociations that occur in aphasic patients.

In sum, the present work is a combination of ideas from computational, psycholinguistic, biological, developmental, and neuropsychological literatures. When a cross-domain approach is taken, solutions for particular problems in one domain propagate and interact with those in other domains. By instantiating these ideas from different domains in the model's input representations and network architecture, and using connectionist-learning algorithms to glue them together, the resulting model, like its human counterparts, has emergent abilities that arise out of the complex interactions among these different systems.

Note

1. A language is not a large, but finite, set of sentences. Rather it is made up of recursive structures that allow for utterances of infinite length (subject to memory constraints). While the present model is constrained in a finite way, its approach to creating language strings would allow it to handle languages with recursive properties. In this model, sentences are produced incrementally, making use of the representations that are activated

at any one moment. This approach to production does not place finite constraints on how long sentences can be. But, at this point, it is not clear whether messages can be controlled in a way that allows for recursive structures.

Acknowledgments

This research formed a part of the author's Ph.D. dissertation at the University of Illinois at Urbana-Champaign. Preparation of this article was supported by National Science Foundation Grant SBR 98-73450, the Language Processing Training Grant T32MH 19990, and the National Institutes of Health Grant DC-00191. I would like to thank Gary Dell for his helpful comments, detailed reviewing of the manuscript, and support throughout this project. I would also like to thank Kathryn Bock, Nick Chater, Victor Ferreira, Cindy Fisher, Adele Goldberg, Zenzi Griffin, Harlan Harris, Dennis Norris, Kris Onishi, Doug Rohde, and two anonymous reviewers for their useful suggestions and comments.

Appendix A

The models were implemented in the LENS 2.3.3 neural network software (Rohde, 1999). The learning algorithm was back-propagation, using a modified momentum algorithm (doug momentum), which is similar to standard momentum descent with the exception that the pre-momentum weight step vector is bounded so that its length cannot exceed 1.0 (Rohde, 1999). Momentum was 0.9. Learning rate started at 0.2 and was reduced linearly until it reached 0.05 at 2,000 epochs, where it was fixed for the rest of training. Batch size was set to be the size of the training set (501). The *cwhere* and *word* units used the soft-max activation function. Soft-max units caused the output to be passed through an exponential function, which magnified small differences, and the result was then normalized (leaving only the most activated unit, and squashing the activation of all the weaker competitors). Because soft-max units were used for the word output units, the error function for these units was the divergence function (sum over all units: target $\times$ log(target/output)). All other units used the logistic activation function.

In all the models, the *event-semantics* units were the only units that provided information about the target sentence order. So, for the dative sentence *A man bake a cake for the cafe*, there were three event semantics units CAUSE, CREATE, and TRANSFER. For the prepositional dative structure, the CAUSE feature would have an activation of 1.0, the CREATE feature would have an activation of 0.8, and the TRANSFER feature would have an activation of 0.64 (80% of 0.8). For the double object dative (e.g., *A man bake the cafe a cake*), the CREATE feature had an activation of 0.64 and the TRANSFER feature had an activation of 0.8.

The *where–what* and *cwhat–cwhere* links instantiated variables that were used to store the message. Before the production of each sentence, the links between the *where* and *what* units were set to 0 initially, and then individual links between *where* roles and *what* units were made by setting the weight to a value of 6. The *cwhat* units had a corresponding link to the *cwhere* units with the same level of weight. The LENS software allowed code to be run before each

sentence sequence was initiated, and so functions that set the value of weights were used to set the message representation before production of a sentence.

The *where*, *what*, and *cwhat* units were also unbiased to make them more input driven (all other units except *context* had bias). The *cwhat* units received the previous timestep activation of the *what* units as targets. Their error function was the cross-entropy function (sum over all units: target $\times$ log(target/output) + (1 − target) $\times$ log((1 − target)/(1 − output))).

To represent temporal information, copies of previous network states are copied into special units, which Rohde (1999) calls *elman* units, and then these units are used as input for the next state of the model (Elman, 1990). The *cwhere2* units are *elman* units that summed their activation from the *cwhere* units and their own previous activation. The *context* units were *elman* units that were initialized to 0.5 at the beginning of a sentence. The *cword* units were also *elman* units that received their values from the sum of external input (representing the previous word in the sequence) and the output of the *word* units. So, during production, the external input would be 0, and the *cword* units would only be a copy of the previous *word* units activation. But during comprehension, the previous *word* unit's activation and the external input were summed. The *cword* and *cwhere2* units were initialized to 0 at the start of each sentence.

To generate a message, a verb was randomly selected from the list of possible verbs. Then arguments were randomly chosen from the possible nouns that that were appropriate for that verb. Also, random selection of adjectives was done within the adjectives that were appropriate for a particular head noun (in terms of animacy of head noun). Prepositions were sometimes selected by the construction (*by* in the passive) or by random selection from those that were appropriate to the construction or the verb. Because messages with a single event role were quickly learned, the training sets were arranged so that the verbs *is*, *dance*, and *sleep* would be less frequent than other verbs. To reduce their occurrence, when these verbs were chosen, the verb selection was repeated. Sometimes the second selection would select other verbs. Only if they occurred in the second selection were they allowed to be the basis for a message. Every message ended with two end of sentence markers, so that if the model changed its sentence structure, it could fully produced longer sentence structures (actives were shorter than passives by two words).

The Dual-path model had 64 *cword* units, 10 *ccompress* units, 52 *cwhat* units, 4 *cwhere* units, 4 *cwhere2* units, 8 *event-semantics* units, 20 *hidden* units, 20 *context* units, 4 *where* units, 52 *what* units, 10 *compress* units, and 64 *word* units. Inadvertently the semantics for two of the prepositions (*under*, *in*) were left out of all the models, but this is unlikely to influence any of the results, because these prepositions were not crucial for any of the tests and comparisons were between models. The prepositions *for*, *to*, *with*, and *by* did not have lexical semantics because they were associated with event semantics or syntactic frames. The verbs *go*, *make*, *give*, *put*, and *fill* were considered light verbs and did not have verb semantics. The No-event-semantics and the Linked-path models had the same number of units in each layer as the Dual-path model. The Prod-SRN model was designed to use the same training/testing patterns as the other models. This was done by placing the static binding-by-space message into the weights between the *bias* unit (an invisible unit that was always on and connected to all biased units) and the *message* units. The Prod-SRN model had 64 *cword* units, 130 *message* units (3 slots with 35 units each and an action slot with 25 units), 50 *hidden* units, 50 *context* units, and 64 *word* units.

References

Baker, C. L. (1979). Syntactic theory and the projection problem. *Linguistic Inquiry*, *10*, 533–581.

Baldwin, D. A. (1993). Early referential understanding: Infants' ability to recognize referential acts for what they are. *Developmental Psychology*, *29*, 832–843.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences*, *22*, 577–660.

Berndt, R. S., Mitchum, C. C., Haendiges, A. N., & Sandson, J. (1997). Verb retrieval in aphasia: 1. Characterizing single word impairments. *Brain & Language*, *56*, 68–106.

Bloom, P., Peterson, M. A., Nadel, L., & Garrett, M. F. (1999). *Language and space*. Cambridge, MA: MIT Press.

Bock, J. K. (1982). Toward a cognitive psychology of syntax: Information processing contributions to sentence formulation. *Psychological Review*, *89*, 1–47.

Bock, J. K. (1986). Meaning, sound, and syntax: Lexical priming in sentence production. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *12*, 575–586.

Bock, J. K. (1987). Co-ordinating words and syntax in speech plans. In A. W. Ellis (Ed.), *Progress in the psychology of language* (Vol. 3, pp. 337–390). London: Erlbaum.

Bock, K., & Levelt, W. (1994). Language production: Grammatical encoding. In M. A. Gernsbacher (Ed.), *Handbook of psycholinguistics* (pp. 945–984). San Diego, CA: Academic Press.

Bock, K., Loebell, H., & Morey, R. (1992). From conceptual roles to structural relations: Bridging the syntactic cleft. *Psychological Review*, *99*, 150–171.

Breedin, S. D., Saffran, E. M., & Schwartz, M. F. (1998). Semantic factors in verb retrieval: An effect of complexity. *Brain and Language*, *63*, 1–31.

Brooks, P. J., & Tomasello, M. (1999). How children constrain their argument structure constructions. *Language*, *75*, 720–738.

Chang, F., Dell, G. S., Bock, K., & Griffin, Z. M. (2000). Structural priming as implicit learning: A comparison of models of sentence production. *Journal of Psycholinguistic Research*, *29*, 217–229.

Chomsky, N. (1957). *Syntactic structures*. The Hague: Mouton.

Christiansen, M. H., & Chater, N. (1999). Towards a connectionist model of recursion in human linguistic performance. *Cognitive Science*, *23*, 157–205.

Clark, E. (1978). Discovering what words can do. In D. Farkas, W. M. Jacobsen, K. W. Todrys (Eds.). Papers from the parasession on the lexicon (pp. 34–57). Chicago, IL: Chicago Linguistics Society.

Clark, E. (1987). The principle of contrast: A constraint on language acquisition. In B. MacWhinney (Ed.), *Mechanisms of language acquisition* (pp. 1–33). Hillsdale, NJ: Lawrence Erlbaum Associates.

Clark, E. V., & Carpenter, K. L. (1989). The notion of source in language acquisition. *Language*, *65*, 1–32.

Dell, G. S. (1986). A spreading-activation theory of retrieval in sentence production. *Psychological Review*, *93*, 283–321.

Dell, G. S., Schwartz, M. F., Martin, N., Saffran, E. M., & Gagnon, D. A. (1997). Lexical access in aphasic and nonaphasic speakers. *Psychological Review*, *104*, 801–838.

Dell, G. S., Chang, F., & Griffin, Z. M. (1999). Connectionist models of language production: Lexical access and grammatical encoding. *Cognitive Science*, *23*, 517–542.

Dowty, D. (1991). Thematic proto-roles and argument selection. *Language*, *67*, 547–619.

Elman, J. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Elman, J. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, *48*, 71–99.

Ferreira, V. S. (1996). Is it better to give than to donate? Syntactic flexibility in language production. *Journal of Memory and Language*, *35*, 724–755.

Ferreira, V. S., & Dell, G. S. (2000). Effect of ambiguity and lexical availability on syntactic and lexical production. *Cognitive Psychology*, *40*, 296–340.

Fisher, C., Gleitman, L. R., & Gleitman, H. (1991). On the semantic content of subcategorization frames. *Cognitive Psychology*, *23*, 331–392.

Fodor, J. A., & Pylyshyn, Z. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, *28*, 3–71.

Garnsey, S. M., Pearlmutter, N. J., Myers, E., & Lotocky, M. A. (1997). The contribution of verb bias and plausibility to the comprehension of temporarily ambiguous sentences. *Journal of Memory and Language, 37*, 58–93.

Garrett, M. F. (1988). Processes in language production. In F. J. Newmeyer (Ed.), *Linguistics: The Cambridge survey, Vol. 3. Language: Psychological and biological aspects* (pp. 69–96). Cambridge, UK: Cambridge University Press.

Goldberg, A. E. (1995). *Constructions: A construction grammar approach to argument structure*. Chicago: University of Chicago Press.

Goodglass, H., & Kaplan, E. (1983). *The assessment of aphasia and related disorders*. Philadelphia, PA: Lea & Febiger.

Gordon, J. K., & Dell, G. S. (2002). Learning to divide the labour between syntax and semantics: A connectionist account of deficits in light and heavy verb production. *Brain and Cognition, 48*, 376–381.

Griffin, Z. M., & Bock, K. (2000). What the eyes say about speaking. *Psychological Science, 11*, 274–279.

Gropen, J., Pinker, S., Hollander, M., Goldberg, R., & Wilson, R. (1989). The learnability and acquisition of the dative alternation in English. *Language, 65*, 203–257.

Gropen, J., Pinker, S., Hollander, M., & Goldberg, R. (1991). Affectedness and direct objects: The role of lexical semantics in the acquisition of verb argument structure. *Cognition, 41*, 153–195.

Hadley, R. L. (2000). Cognition and the computational power of connectionist networks. *Connection Science, 12*, 95–110.

Hertz, J., Krogh, A., & Palmer, R. G. (1991). *Introduction to the theory of neural computation* (Vol. 1). Redwood City, CA: Addison-Wesley.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review, 104*, 427–466.

Jackendoff, R. (1990). *Semantic structures*. Cambridge, MA: MIT Press.

Jacobs, R. A., Jordan, M. I., & Barto, A. G. (1991). Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks. *Cognitive Science, 15*, 219–250.

Jordan, M. (1986). *Serial order: A parallel distributed processing approach*. ICS Technical Report No. 8604, University of California at San Diego, La Jolla, CA.

Karnath, H. (2001). New insights into the functions of the superior temporal cortex. *Nature Reviews Neuroscience, 2*, 568–576.

Kosslyn, S. (1980). *Image and mind*. Cambridge, MA: Harvard University Press.

Lakoff, G. (1987). *Women, fire, and dangerous things*. Chicago: University of Chicago Press.

Landau, B., & Jackendoff, R. (1993). "What" and "where" in spatial language and spatial cognition. *Behavioral and Brain Sciences, 16*, 217–238.

Langacker, R. (1987). *Foundations of cognitive grammar*. Stanford, CA: Stanford University Press.

Levelt, W. J. M., Roelofs, A., & Meyer, A. S. (1999). A theory of lexical access in speech production. *Behavioral and Brain Sciences, 22*, 1–38.

MacDonald, M. C., Pearlmutter, N. J., & Seidenberg, M. S. (1994). The lexical nature of syntactic ambiguity resolution. *Psychological Review, 89*, 483–506.

MacWhinney, B. (1987). The competition model. In B. MacWhinney (Ed.), *Mechanisms of language acquisition* (pp. 249–308). Hillsdale, NJ: Lawrence Erlbaum Associates.

Mandler, J. M. (1992). How to build a baby: II. Conceptual primitives. *Psychological Review, 99*, 587–604.

Marcus, G. F. (1998). Rethinking eliminative connectionism. *Cognitive Psychology, 37*, 243–282.

Marcus, G. F. (2001). *The algebraic mind*. Cambridge, MA: MIT Press.

McClelland, J. L., & Kawamoto, A. H. (1986). Mechanisms of sentence processing: Assigning roles to constituents of sentences. In J. L. McClelland & D. E. Rumelhart (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (pp. 272–325). Cambridge, MA: MIT Press.

Milner, A. D., & Goodale, M. A. (1995). *The visual brain in action*. Oxford: Oxford University Press.

Mishkin, M., & Ungerleider, L. G. (1982). Contribution of striate inputs to the visuospatial functions of parieto-preoccipital cortex in monkeys. *Behavioral Brain Research, 6*, 57–77.

Pinker, S. (1989). *Learnability and cognition: The acquisition of argument structure*. Cambridge, MA: MIT Press.

Pinker, S., & Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition, 28*, 73–193.

Plaut, D. C., McClelland, J. L., Seidenberg, M. S., & Patterson, K. (1996). Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychological Review, 103*, 56–115.

Plunkett, K., & Juola, P. (1999). A connectionist model of English past tense and plural morphology. *Cognitive Science*, *23*, 464–490.

Potter, M. C., & Lombardi, L. (1990). Regeneration in the short-term recall of sentences. *Journal of Memory and Language*, *29*, 713–733.

Regier, T. (1995). A model of the human capacity for categorizing spatial relations. *Cognitive Linguistics*, *6*, 63–88.

Rohde, D. L. T. (1999). *LENS: The light, efficient, network simulator*. Carnegie Mellon University, Department of Computer Science, Pittsburgh, PA.

Rohde, D. L. T., & Plaut, D. C. (1999). Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition*, *72*, 67–109.

Rumelhart, D. E., & McClelland, J. L. (1986). On learning the past tenses of English verbs. In J. L. McClelland, D. E. Rumelhart, & The PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition: Vol. 2. Psychological and biological models* (pp. 216–271). Cambridge, MA: MIT Press.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

Shastri, L., & Ajjanagadde, V. (1993). From simple association to systematic reasoning: A connectionist representation of rules, variables, and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences*, *16*, 417–494.

St. John, M. F., & McClelland, J. L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence*, *46*, 217–257.

Talmy, L. (1999). Fictive motion in language and ception. In P. Bloom, M. A. Peterson, L. Nadel, & M. F. Garrett (Eds.), *Language and space* (pp. 211–276). Cambridge, MA: MIT Press.

Tomasello, M. (1999). *The cultural origins of human cognition*. Cambridge, MA: Harvard University Press.

Tomasello, M., Akhtar, N., Dodson, K., & Renau, L. (1997). Differential productivity in young children's use of nouns and verbs. *Journal of Child Language*, *24*, 373–387.

Ullman, M. T. (2001). A neurocognitive perspective on language: The declarative/procedural model. *Nature Reviews Neuroscience*, *2*, 717–726.