Syntactic Systematicity Arising from Semantic Predictions in a Hebbian-Competitive Network

Robert F. Hadley
Adam Rotaru-Varga
Dirk V. Arnold
Vlad C. Cardei

School of Computing Science
Simon Fraser University
Burnaby, B.C., V5A 1S6
Canada
Phone: 604-291-4488, Fax: 604-291-3045
hadley@cs.sfu.ca, arotaru@cs.sfu.ca, dirk@cs.sfu.ca, vcardei@cs.sfu.ca

Running Title: Systematicity from Semantic Predictability

ABSTRACT

A Hebbian-inspired, competitive network is presented which learns to predict the typical semantic features of denoting terms in simple and moderately complex sentences. In addition, the network learns to predict the appearance of syntactically key words, such as prepositions and relative pronouns. Importantly, as a by-product of the network's semantic training, a strong form of syntactic systematicity emerges. This systematicity is exhibited even at a novel, deeper level of clausal embedding. All network training is unsupervised with respect to error feedback. A novel variant of competitive learning, and an unusual hierarchical architecture are presented. The relationship of this work to issues raised by Marcus (1998) and Phillips (2000) is explored.

Keywords: Systematicity, Semantic Features, Language Acquisition, Competitive Learning, Connectionism.

1. Introduction[0]

Within the last decade, simple recurrent networks (SRNs) have attracted considerable notice among connectionists involved in language acquisition research. In no small measure, this attention is due to the work of Elman (1990, 1993), and those who have adopted his "temporal sequencing" strategy (for examples, see Christiansen & Chater, 1994; Christiansen, Chater, & Seidenberg, 1999). However, despite a number of intriguing results, reservations about the applicability of SRNs, within the domain of language learning, persist. Some of these misgivings concern the range of syntactic patterns (Marcus, 1998; Rispoli, 1999) that SRN's can actually master. Other doubts involve the degree of systematicity (or word-position generalization) that SRNs can attain (see Hadley, 1994a, 1994b, in press; Marcus, 1998; Phillips, 2000). Still other qualms concern the nature of the learning algorithm employed in most SRN experiments, viz., the backpropagation of error. Indeed, Elman and co-authors have expressed some misgivings about employing a training algorithm whose biological basis has persistently eluded discovery (Elman et al, 1996).

In this paper, we shall have little to say about the first class of worries just mentioned (i.e., the scope of learnable patterns). Rather, our discussion and results will address the last two areas of concern. We hope to show that a shift from local, lexical representations to distributed, semantic representations can do much to allay problems with systematicity. Moreover, we present a Hebbian-competitive network model, which exhibits a good

[0]Robert Hadley gratefully acknowledges the support of the Natural Sciences and Engineering Research Council of Canada during the period of this research.

approximation to the SRNs predictive behaviour, while relying upon biologically inspired training methods. However, before delving into the details of our own model, it will be helpful to consider some of the issues surrounding systematicity.

Between 1990 and 1994, substantial research was directed at demonstrating the capacity of SRNs to predict the syntactic category of the next word in a sentence, given some current word of input (cf. Elman, 1990, 1993; Christiansen & Chater, 1994). In addition, since about 1990, a number of publications have described networks which achieve some degree of syntactic or semantic *systematicity* (cf. Chalmers, 1990; Christiansen and Chater, 1994; Elman, 1998; Hadley, 1994a, 1994b; Hadley & Hayward, 1997; Hadley & Cardei, 1999; Niklasson and van Gelder, 1994; Phillips, 1994). Overwhelmingly, this research on systematicity has focused on the capacity of connectionist networks to generalize the *use or interpretation* of terms to novel syntactic positions within sentences. However, as explained in (Hadley, 1994b and Hadley, in press), the forms of systematicity achieved thus far by SRNs lack robustness – they display systematicity in very limited contexts, and only for a small fraction of the words, or symbols, involved. Admittedly, Elman has demonstrated (1998), that with prolonged training on a single small corpus, an SRN can generalize its predictions for a given noun, hitherto seen only in the position of grammatical subject, to the position of grammatical object. However, further investigation of this phenomenon by Voll & Hadley (in preparation) demonstrates that this capacity to generalize predictions is indeed fragile. When a training corpus is based upon a modestly *non-deterministic* grammar (i.e., one that permits alternative grammatical categories to appear directly after a given context) a backpropagation-trained SRN will often fail to generalize into the coming "permissive context". For example, our SRN could not predict the appearance of hitherto, positionally-restricted nouns in a context where *either* a noun or a verb could appear (e.g., following a relative pronoun).

Now, it is noteworthy that there do exist connectionist networks capable of displaying robust forms of systematicity. Indeed, the networks described in Hadley & Hayward (1997) and Hadley & Cardei (1999) each satisfy a stringent definition of *strong semantic systematicity*. This definition not only demands a strong form of generalization, but requires the given agent or network to assign (following training) precise, appropriate meanings to a wide variety of representative test sentences. However, as explained in (Hadley, in press), all networks currently known to satisfy this stringent definition employ *classical*, combinatorially predisposed, representational layers. A notable aspect of SRNs is that they do not presuppose the presence of combinatorially-endowed wiring. In the work presented here, we have retained this latter quality.

In what follows, we present a Hebbian-competitive network which satisfies a definition of *strong systematicity* first offered in (Hadley, 1994b). Briefly stated, that definition requires that an agent *learn* to generalize the use of a *significant fraction* of its vocabulary to novel syntactic positions. These novel positions (i.e., novel to the words whose usage is being generalized) must occur not only in simple sentences, but also in embedded clauses. In this regard, a word or symbol is considered to occupy a novel position (e.g., grammatical subject) only if the agent has not encountered that word in that syntactic position at any level of sentential embedding. Significantly, the present system achieves this property *without presupposing* the existence of previously acquired, classical semantic representations. (We wish to stress, however, that our system is not designed to satisfy *all* forms of strong generalization/systematicity that have been discussed in recent years. See this footnote.[1])

[1] Quite apart from the notion of *strong semantic systematicity* examined in Hadley & Hayward (1997), Marcus (1998) and Phillips (2000) have posed challenges involving cases of human generalization which presuppose considerable prior linguistic competence and high-level cognitive skills on the part of the human involved. These challenges have much in common with cases of virtually instantenous rule acquisition examined in Hadley (1993). We return to these issues in our general discussion (see section 10).

By contrast with the SRN based experiments cited above, a major task of our network is to predict the *semantic* category of the next word in a sentence, rather than its syntactic category. Our belief is that, to a large degree, systematicity at the syntactic level derives from predictability at a semantic level. For this reason, both the input and output layers of our network are used to encode distributed semantic (feature) representations, rather than local lexical representations. Of course, we recognize that the use of distributed, feature representations (at the input and output levels) may also improve the systematicity of standard SRNs, trained by backpropagation. However, it is doubtful whether backprop-trained SRNs could match the degree of generalization and precision attained by the competitive model presented here, for the following reason: the novel competitive algorithm employed here ensures that input semantic vectors, which may share as few as 50% features in common, can be mapped to precisely the same category within the network's first hidden layer. Moreover, depending on the chosen parameter settings, the network can assign input feature vectors to one of N, *precise* sub-categories.

Apart from the advantage just cited, our choice of a Hebbian-competitive model reflects our second major goal, which is, to demonstrate one way in which a useful approximation to SRN behaviour can arise through purely self-organizing forms of connectionist learning. We do not suggest that our competitive network represents a straightforward *reduction* of a backpropagation-trained SRN to a self-organizing network. The differences mentioned in the preceding paragraph would preclude that possibility. However, we believe that the model presented here constitutes something strongly akin to a competitive implementation of SRNs, and, as mentioned, possesses some advantages over standard SRNs. Prominent connectionist researchers (including Elman et al, 1996) have remarked upon the desirability of replacing backpropagation-based networks with behaviourally similar architectures which are (at least) closer to biological reality. We believe that the model described here represents an appreciable step in this direction.

In addition, we have taken other steps in the direction of cognitive plausibility (though we are well aware that much remains to be done in this regard). For example, we have employed comparatively sparse sets of training data. Pace Elman, *relative* sparsity is an important constraint to satisfy.[2] During training, the network is exposed to 3000 distinct sentences, while the set of *potential* test sentences numbers well over 300 million. In addition, we have ensured that training and test data are generated from a non-deterministic grammar. Both subject-relative and object-relative clauses are permitted to occur within the same sentence. Relative clauses and prepositional phrases may likewise occur in the same sentence.

Furthermore, during training, *two-thirds* of all nouns are restricted to a single syntactic position and clausal embedding is restricted to a depth of two. During testing, these restrictions are dropped. The network successfully predicts the appearance of hitherto restricted nouns in all grammatically permissible contexts. This includes occurrences in relative clauses of depth three, and in (non-deterministic) contexts, where either a noun or verb might be expected.

Having said this, we wish to emphasize that the network described here is *not* intended as a general mechanism for language acquisition. Clearly, there is much more to acquisiton than the mere capacity to predict semantic categories of successive words in a sentence. However, we believe that something akin to the model presented here could be one aspect of a larger learning system within humans. In Hadley and Cardei (1999), it was shown that the presence of a sub-network, which supplies *state information* about the current semantic context, can play a key role in the success of a larger

[2]Elman et al (1996) contend that the data children encounter during language acquisition is far less impoverished than alleged by conventional (Chomsky inspired) psycholinguistic wisdom. In terms of sheer numbers of sentences encountered, we do not dispute this claim. Nevertheless, it remains true that the sentential data children encounter is very *sparse relative to* the entire set of humanly comprehendable sentences in their mother tongue.

S $\rightarrow$ NP V NP

NP $\rightarrow$ N | N RC | N PP

N $\rightarrow$ women | girls | birds | bats | men | boys | chairs
| baseballs | dogs | tables | cats | mice

V $\rightarrow$ chase | sees | swing | love | avoid | follow | bump
| hit | consume | dislike

RC $\rightarrow$ that V NP

RC $\rightarrow$ that N V

PP $\rightarrow$ Prep NP

Prep $\rightarrow$ from | with

Figure 1: The grammar for generating training and test sentences.

language acquisition network. The model described herein consititutes a substantial improvement on the self-organizing map, which played a "semantic state" indicating role in the Hadley-Cardei paper.

2. Task Overview.

The connectionist network presented here is designed to process words, taken in sequence, from a variety of sentences generated according to the syntax displayed in Figure 1.

All words shown in Fig. 1 have previously been assigned semantic feature vectors. The assignment of semantic features is displayed in appendix A. Less than half of the possible "candidate" features are assigned to any given noun or verb, since many pairs of candidates are semantically inconsistent.

Now, although there are similarities with Elman's well known models (1990), the *primary* task of our network (shown in Figure 2) is not to predict *words* or syntactic categories. Rather, its main task is to predict typical semantic features for the next word in a sentence, when given semantic features for the current word.

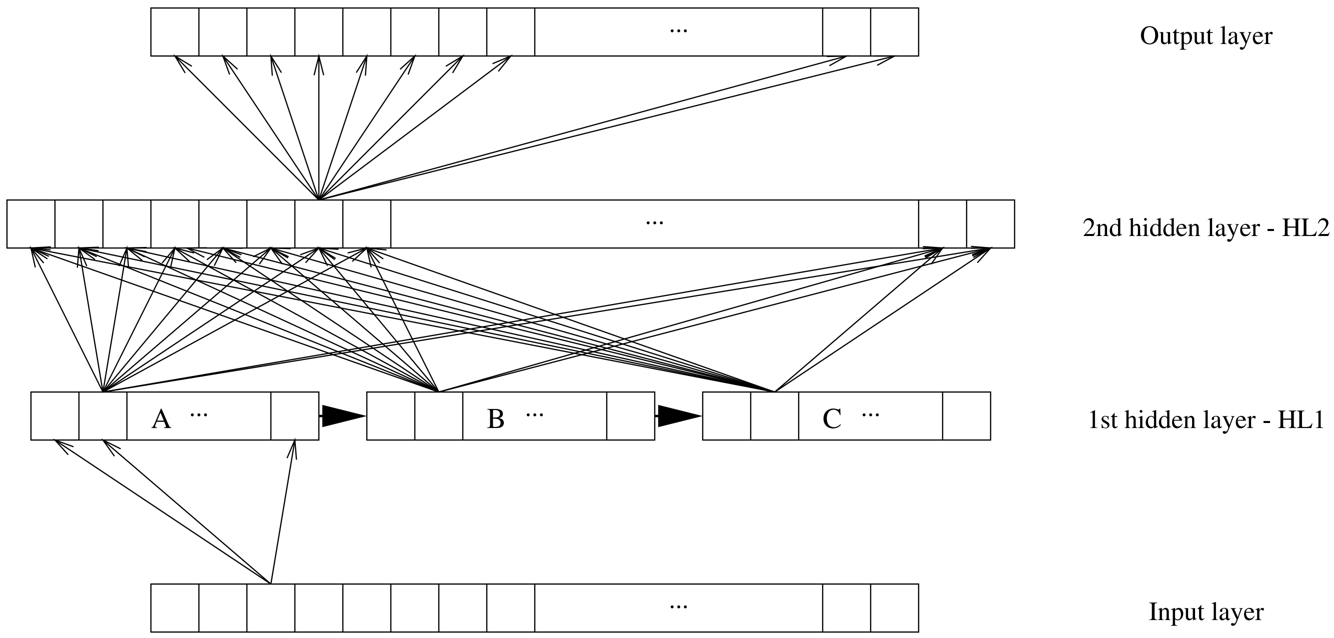

Figure 2: Overall Architecture of the Hebbian-Competitive network. Regions B and C of the first hidden layer only serve as memory copies, and are *not* connected to the input layer.

In order to challenge the generalization capacity of our network, our training corpus has been designed to ensure that 2/3 of all nouns are limited to a single syntactic position *during training*. Each sentence selected from the training corpus is presented, one word at a time, for processing by the network. In effect, this means that the word's semantic feature vector is activated within the network's input layer. (See Figure 2). To enable the system to learn to predict *typical* semantic features, semantic information for the *next* word is presented to the output layer's semantic region. This requires that we assume that a learning agent has access not only to the current input word, but to the next word to appear. This assumption is shared by Elman's prediction models (1990, 1993) and by several other systems inspired by Elman's work. One rationale for this approach is that the learning agent could wait to hear the next word before attempting to learn anything from the current word.

3. General Overview of Architecture and Processing.

As apparent in Fig. 2, the network contains four separate layers, namely: an input layer, a first hidden layer (HL1), a second hidden layer (HL2), and an output layer.

The input and output layers each contain 51 units, thereby enabling the representation of the 51-element input/output vectors. HL1 is somewhat unusual in that it contains three distinct regions, which are named (left to right) region A, region B, and region C. Links between the input layer and region A of HL1 are fully connected. Moreover, *each region* of HL1 is fully connected to HL2, which in turn is fully connected to the output layer. Region A of HL1 is a winner-take-all (WTA) competitive cluster, containing 20 nodes. Regions B and C each contain 20 nodes, and serve as memory copies of the prior, successive contents of region A. Hence, region A is linked to region B by 1 to 1 mapping links (connecting corresponding nodes). A similar set of links connects region B to region C.

The second hidden layer (HL2) is a single competitive (WTA) cluster, containing 400 nodes. HL2 receives input from all three regions of HL1. Links entering HL2, like those entering region A of HL1, are trained via a novel competitive algorithm, described in section 7. In contrast, links connecting HL2 to the output layer are trained via a simple, non-competitive Hebbian (Hebb, 1949) method (section 7).

The post-training role of region A, in HL1, is to categorize the various input vectors into semantic groups, corresponding to typical nouns, typical verbs, etc. The role of HL2, on the other hand, is to recognize ternary patterns that appear in the three regions of HL1. Thus, HL2 is a higher-order pattern recognizer.

In a given training cycle, a 51-element feature vector, representing the semantics of the current input word, is presented to the input layer. Concurrently, another feature vector, belonging to the *next* input word, is activated within the output layer. Following that, activation spreads from the input level to region A, where competition ensues and a single winner is selected. Once selected, weight modification applies to links entering that winning node, in accord with the algorithm given in section 7. Moreover, the winner then fires, and activation propagates along two sets of links, specifically, the links flowing upwards (to HL2) and those spreading to region B in HL1.

Consider the former case, viz., the spread of activation from HL1 to HL2. Once a winner is selected in region A, activation immediately flows upwards from all three regions of HL1 to HL2. Note that the pattern of activation reaching HL2 reflects region A's winner, and also the contents of B and C at the time this winner was selected. Once activity reaches HL2, competition again ensues and a single HL2 winner is selected. Weight modification applies to links entering this HL2 winning node, again in accordance with the algorithm presented in section 7.

Now consider the second case, viz., activity flow between HL1 regions. Links connecting regions within HL1 are non-trainable and have weights of +1. Once region A's winner is selected and fires, activation reaching region B from A will trigger a chain reaction, causing the *prior* activation pattern within B to be spread to C. Region B then assumes the pattern which it has just received from A and C assumes the pattern arriving from B. (Our assumption here is that the functionality of regions B and C differs from that of A – the latter being a competitive cluster.) Note that, since region A is a WTA cluster, and because regions B and C simply retain successive (memory) copies of region A, at most one node will be active within each region at any given time.

Let us finally consider the topmost set of links – those between HL2 and the output layer. Once a winner is selected within HL2, a simple Hebbian incrementing of weights occurs on just those links that flow from this winner to all currently active nodes within the output layer. (Recall that a subset of output nodes was made active when the semantic vector for the next word was presented to the output layer at the beginning of the present cycle).

4. Input/Output Feature Vectors

The 51-element feature vectors, representable within the input and output layers, contain 35 ostensibly semantic features, and an additional 16 features whose interpretation is less straightforward. Units within the input layer assume only binary values – 1 if the feature is present and zero otherwise. During training, the output layer likewise assumes binary values, but during testing, propagation of activity from lower levels invariably causes output units to assume real-number values between zero and one.

Features assigned to the 12 nouns are taken to be subsets of the following: *animate, inanimate, two-legs, four-legs, talks, barks, meows, squeaks, has-weight, has-size, has-shape, has-location, furry, large, small, heavy, light, laughs, bites, long-snout, flat-face, small-nose, rigid, flexible, tubular, round.*

All nouns employed here denote physical objects, and for this reason, the concepts expressed by these nouns share certain features: viz., has-weight, has-size, has-shape, has-location.

In contrast, features assigned to the 11 verbs typify either actions or other relationships, and are taken to be subsets of the following: *physical-motion, involves-contact, involves-animate, rapid, slow, emotive, feeling-nice, feeling-bad, involves-perceiving.*

Semantic features adopted here are admittedly incomplete and approximate. However, they serve to convey the general approach we have adopted. In the case of some noun-verb pairs, a feature is shared by both a noun and a verb. Such overlap in feature representation does not and would not affect the formation of separate "noun feature" and "verb feature" recognizers within layer HL1, provided the overlap between noun features and verb features is small (less than 25%).

Further details on the assignment of specific features to nouns and verbs are presented in appendix A.

As mentioned a few paragraphs back, 16 of the 51 elements within the input/output vectors are devoted to features whose interpretation is not clearly semantic. These 16 features are divided equally among the following four terms: 'that' (the relative pronoun), 'with' and 'from' (the prepositions), and the period ('.'). While it is arguable that humans do, at some point, assign semantic information to 'that', 'with', and 'from', it is doubtful whether they have done so in the early "semantic bootstrapping" phases of acquisition. Moreover, the role of the period seems more clearly syntactic than semantic. In its case at least, we construe the assignment of four features (out of 51) as a form of syntactic representation.[3] With regard to the relative pronoun and prepositions just quoted, we favor a lexical (or perhaps phonetic) construal. We suggest that in the (plausible) early absence of known semantic information for each of these words, the learning agent may fall back on some form of lexical/phonetic encoding. For convenience, we have arbitrarily assigned to each of these three lexical items a set of four features (see Appendix A). The choice of four features (as opposed to 12, say) merely reflects concern for efficiency in our simulation. Nouns and verbs were respectively assigned 26 and 9 features. Yet, these numbers did not diminish the network's capacity to develop separate recognition nodes (in HL1) for noun and verb vectors. We assigned fewer features to verbs than to nouns simply because we found it easier to think of noun features.

5. The Training Corpus

[3]We have followed Elman in including periods within the input stream. Periods may correspond to extended silences in spoken speech, and in "motherese" we do find such silences (O'Grady, 1997).

A corpus of 3000 sentences was generated in accordance with the grammar displayed in Figure 1. This generation process was subject to several constraints, to ensure that ample opportunity for generalization would arise.

The constraints include:

- Four of twelve nouns (namely, *tables, bats, chairs, mice*) were precluded from appearing as grammatical subjects during training. Hereafter, we refer to these as non-subject (NS) nouns.
- Four of twelve nouns (namely, *boys, girls, dogs, women*) were precluded from appearing as direct objects during training. These are dubbed NDO nouns (non-direct object).
- The remaining four nouns (*baseballs, birds, cats, men*) were unrestricted (modulo grammaticality). Of these last four nouns, only 'baseball' possesses the feature 'inanimate'. This is the only inanimate noun ever to occur, *during training*, in the position of subject. Moreover, it was permitted to occur only in conjunction with the verbs 'bump' and 'hit'. Significantly, 'baseballs' shares at most two-thirds of its features with any of the NS nouns.
- 25% of all sentences generated contained one or two relative clauses (RC). Of this 25%, half contained just a single RC. The remaining half contained two consecutive RCs, taking the form of (1) or (2) below. Forms (1) and (2) occur with equal frequency.
    1. *that Verb Noun that Verb Noun*, as in "that chase cats that see mice".
    2. *that Verb Noun that Noun Verb*, as in "that chase cats that boys see".
- An additional 25% of sentences contain a single prepositional phrase.

- The remaining 50% of sentences are of the simplest form (Noun Verb Noun).

Apart from constraints listed above, each training sentence is generated in a fashion which maximizes randomness. That is, subject to the constraints, whenever a decision is made about which word to pick, or whether to employ a simple or complex sentence, a random selection is made.

6. Initialization of Weights.

Prior to the onset of all training, weights on links entering the *competitive clusters* (i.e., region A of HL1 and all of HL2) are initialized as follows: each link entering a given node within these regions is assigned a random weight from the closed interval [0,1]. Then, as customary, weights are normalized to ensure that the sum of weights for each competitive node is one.

In contrast, recall that links entering the output layer are not trained competitively, but by simple Hebbian incrementation. For this reason, all weights assigned to these Hebbian links are initially zero.

7. Training Regime and Algorithms

During the training process, 15,000 sentences are selected (one by one, and in a random fashion) from the 3000 sentence training corpus. On average, then, each training sentence is selected for presentation five times, though in actuality it could happen that some sentences are never selected. This random aspect adds additional realism to the training regime.

As each feature vector (belonging to each word in the input stream) is presented to the lowest layer, activation propagates in the fashion described in section 3 above. Recall that links entering the two memory regions (B and C) are of fixed weight, and serve to permit the sequential transmission of copies from region A.

However, once activation has spread along a set of links entering a competitive cluster, winning nodes are selected according to a formula detailed below. To our knowledge, this formula is novel in its essentials.

7.1 *Winner Selection.*

For all competitive nodes, we distinguish between the initial (raw) activation (here designated as *Received*) and a secondary (refined) activation. In the case of all competitive nodes, whether in HL1 or HL2, 'Received' is simply the standard weighted sum of activation flowing into any given node[4]

Once the *Received* value is obtained for each node j within a given competitive cluster, we compute *j's* refined activation, V_j, as follows:

$$V_j = c * Received_j \ + \ \{(1 \ - \ c) * (Received_j / Max_j)\}$$

where Max_j refers to the maximum previous "$Received_j$" activation for that node, and c is a constant whose value can be experimentally varied. In our implementation, *c=.9* for nodes within HL1 and *c=.1* for nodes in HL2.

Once V_j has been computed for all nodes in a given competitive cluster, the chosen winner is simply that node whose V value is highest. Thus, nodes compete on the basis of their *refined* activation rather than their initially received activations. As customarily occurs in competitive learning, a winning node assumes a value of +1, and losing nodes fall to zero activation.

Before proceeding, it would be well to explain the design considerations which motivated the "winner selection rule" described above. First note that the constant c can,

[4]That is, *Received* of $node_j$ is $\sum_i \omega_{ij} a_i$, where i indexes all nodes providing input to $node_j$, and a_i is the activation of each node so indexed.

depending upon its relative size, either bias the winner selection in terms of the simple Received activation, or in terms of the ratio of Received to the previous maximum for the given node. When c is nearly $+1$, as occurs for HL1 nodes, then winners are selected primarily on the basis of the simple Received activation (since the ratio factor has little effect in this case). Thus, in such cases, winner selection approximates the widespread practice of selecting winners purely on the basis of received activation. This practice favors the tendency for several overlapping input patterns to be recognized by the same winning node within the competitive layer.

In contrast, when c is close to zero, as occurs with HL2 nodes, the bias is placed on the ratio of current Received to the previously received maximum for the given node. Giving preference to this ratio promotes the tendency for a 1-1 mapping to occur between distinct input patterns and distinct winner (recognition) nodes. Since each triplet of activation occurring within the three regions of HL1 represents a distinct syntactic pattern found within input sentences, we wanted to favor 1-1 mappings of this kind. For HL2, the chosen value for c (and values as large as .2) does in fact yield 1-1 mappings *for the training corpus*. Interestingly, values of c above zero yield better results than values of zero for most learning rates.

7.2 *Competitive Weight Modification.*

Weights leading into a chosen winning node are modified according to a familiar Hebbian-inspired, competitive learning algorithm, developed by von der Malsburg (1973). The precise equation for modifying weights on links entering HL1 is:

$$\Delta\omega_{ij} = \eta_0 \left(\frac{a_i}{\sum_k a_k} - \omega_{ij} \right),$$

where j indexes nodes in region A of HL1, i and k each index nodes in the layer just below region A, ω_{ij} denotes the weight from node i to node j, and $\eta_0 = 0.1$ is the learning rate. Recall that activations of nodes in the input layer are either zero or +1.

Weight modification for links connecting HL1 to HL2 is performed in a precisely similar fashion, with the sole exception that the learning rate for HL2 is .001 rather than .1.

7.3 *Modifying Weights Entering the Output Layer.*

Weights connecting HL2 and the output layer are modified by a simple Hebbian process. Recall that, at the time a winner is selected within HL2, an activation vector for the *next* input word is present within the output layer. Thus, several nodes will be active within the output layer. Our Hebbian rule simply increments weights on each link connecting the chosen winner within HL2 with any active node in the output layer. The increment applied is 2.5×10^{-5}.

8. The Test Corpus

A single 3000 sentence test corpus was generated using the grammar previously discussed. With regard to NS and NDO nouns, all previous restrictions were relaxed. That is, every noun could now freely serve as subject or direct object. The overall percentages of sentences containing relative clauses and prepositional phrases was comparable to that found in the training corpus. However, in order to test the network's capacity to generalize to deeper levels of embedding, we ensured that 7% of all sentences contained three consecutive relative clauses. The precise breakdown on sentences containing RCs is as follows:

7% of sentences contain three RCs.

8% of sentences contain exactly two RCs.

10% of sentences contain just one RC.

An additional 25% of sentences contained exactly one prepositional phrase. The remaining 50% of sentences were of the simplest form (N V N). As with the training corpus, subject to the percentages just mentioned, random selections for word choice and clausal-inclusions were made wherever possible.

9. Test Phase and Analysis of Results

The testing procedure involved feeding each of the 3000 test sentences, word by word, through the network. As each word was processed, its semantic features were activated within the input layer. Once activation had propagated to the topmost level, activation vectors appearing within the output layer were accumulated for later averaging. Various cosine comparisons were made between average vectors of predicted words and average vectors of words actually appearing. These comparisons are discussed below and are displayed in Figures 3 through 10. As anticipated, our analyses revealed that the network does learn to predict typical semantic features of words that immediately follow a given word of input.

Significantly, abundant evidence for syntactic systematicity emerged in terms of the network's capacity to process nouns (construed grammatically) in novel positions both within simple sentences and at novel levels of embedding. For example, during the test phase, we compared the average of semantic vectors *predicted* by the "restricted nouns" (those whose position was restricted during training) to the average of semantic vectors that *actually belonged* to words immediately following those given nouns. In detail, we performed a set of six distinct training and test sessions. For each session, cosines were computed for the categories of NS, NDO, and unrestricted nouns (see results in Table I).

Consider for the moment just the *worst* cosines obtained for those six sessions. For the NS noun class, the worst cosine (of the angle between the average of the predicted vectors and the average of vectors actually presented) was .9692, indicating that the vectors were close to each other. The worst cosine value for NDO nouns was .9676, again revealing that the vectors were close. For the class of nouns whose positions were *unrestricted* during training, the result was scarcely greater. In this case, the worst cosine was .9685. All the above cosine values are very good, and the predictive behaviours of both restricted and unrestricted nouns are more than acceptable.

Session	*All*	*NDO*	*NS*	*Unrestricted*
1	.98079	.96763	.98496	.98396
2	.97693	.97104	.98577	.97016
3	.98158	.97633	.98921	.97545
4	.98014	.97880	.98213	.97785
5	.97553	.96942	.98481	.96855
6	.98758	.97671	.96923	.98729
min.	.97553	.96763	.96923	.96855
avg.	.980352	.97446	.98223	.97586
max.	.98758	.9788	.98921	.98729
std.dev.	.004934851	.004753234	.007648992	.007765134

Table I. Cosines for Six Distinct Training-Test Sessions:
All Nouns, NDO nouns, NS nouns, and Unrestricted nouns.

It is relevant to note that differences in cosine values among NS, NDO, and unrestricted nouns are *not* caused by any putative differences in the actual predictions made by nouns in these three classes. Rather, the differences derive from the fact that, within the test corpus, word categories which immediately follow nouns are not entirely uniformly distributed. This is due to weak semantic constraints which were imposed both on the training and test corpora. (For example, only animate nouns could precede the verbs 'see' and 'consume').

During the test phase, and even during most of the training phase, input vectors belonging to *all* nouns select the same, single, winner node within region A of HL1. Consequently, during testing, activation patterns presented to HL2 by the three HL1 regions do *not* reflect variations in the features or identity of any of the nouns involved. For this very reason, for layers above HL1, all nouns in our grammar possess equivalent predictive functionality once training is complete. Thus, other things being equal, post-training systematicity is inevitable.

With regard to word classes other than nouns, only verb input vectors are divided into subclasses by winning nodes in region A. Two separate verb classes are formed, one of which contains verbs possessing the features of mental states, while the other contains verbs pertaining to physical motion.

9.1 What Categories Are Predicted?.

In the remainder of this section we present graphs which illustrate in detail that *input* vectors belonging to *each* syntactic category (noun, verb, preposition, relative pronoun and the 'period') made appropriate predictions for semantic features belonging to words which could legally follow the vector making the prediction. In addition, we examine the kinds of predictions made by differing *sequences* of input categories.

Figure 3 displays those words possessing semantic features that are strongly predicted when the input vector belongs to the category 'noun'. In detail, the height of each bar in the graph displays, in terms of cosines, how closely the semantic feature vector of each vocabulary word matches the *average* vector produced within the output layer when the input word is a noun. As can be seen, all words predicted with any appreciable degree of strength are the very words we would expect, given the grammar involved. All verbs are strongly predicted, as is the period. The prepositions ('with' and 'from') are somewhat less strongly predicted, but with a strength equal to each other. This was expected, and

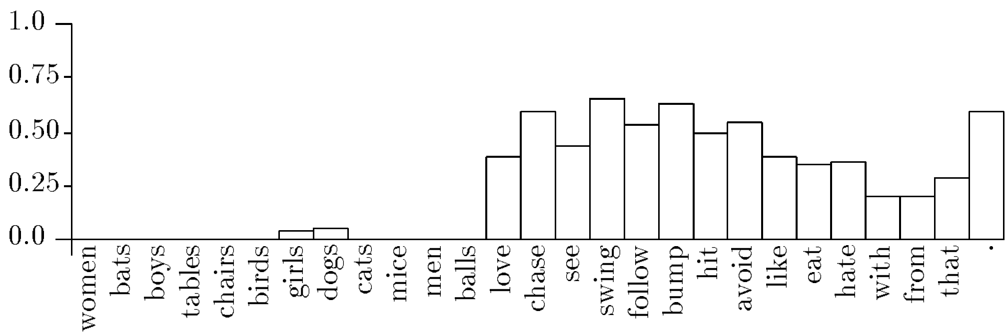

Figure 3: Semantic predictions when the current input is a noun.

these prepositions do occur with equal frequency in both the training and test corpora. The relative pronoun ('that') is predicted more strongly than either of the prepositions, reflecting the fact that relative clauses do frequently follow nouns in both the training and test sets. Note also that very weak predictions for two specific nouns ('girls' and 'dogs') can be discerned. These predictions reflect the fact that, for experimental purposes, we assigned a single verb feature to each of these two nouns. The feature vectors of these two nouns, therefore, do possess a small degree of overlap with feature vectors of verbs (the latter being correctly predicted). Recall, however, that both 'girls' and 'dogs' are correctly classed as *nouns* by nodes in HL1.

Figure 4 is analogous to Figure 3, except that the present graph indicates predictions made when the current input word is a *verb*. As hoped, nouns are strongly predicted to follow the appearance of a verb. This holds not only for the unrestricted nouns, but for the hitherto NS and NDO nouns as well. Thus, we have independent confirmation of the *syntactic* systematicity acquired by the network.

Interestingly, Figure 4 also contains some barely perceptible bars, suggesting that certain verbs and the period are weakly predicted to appear when the current input is a verb. This is actually compatible with the training and test data. Verbs can follow verbs in sentences such as "Girls that boys love like cats". Periods can follow verbs in

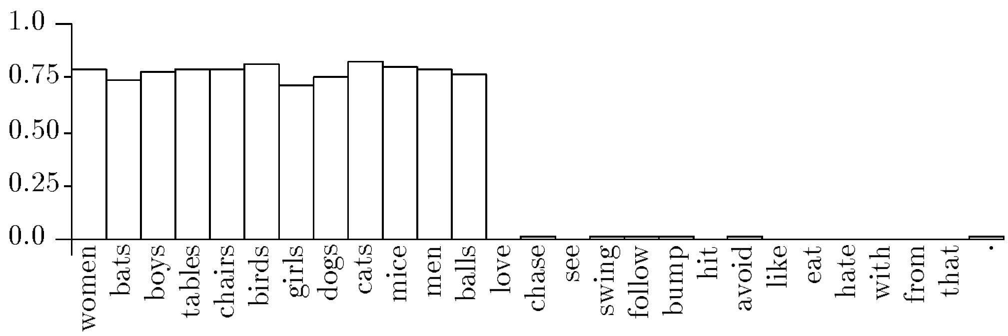

Figure 4: Semantic predictions when the current input is a verb.

sentences such as "Birds see mice that cats chase." However, sentences of both these types are comparatively infrequent in the training corpus, and the feeble (apparent) predictions for verbs and periods might be construed as noise introduced early in the training phase. (Indeed, noise of this kind may be seen in Figure 10.)

In order to discover whether the weakness of the above verb & period predictions is merely due to the "swamping" effects of the strongly predicted noun vectors, we examined a relevant subset of word sequences occurring in the test data. In particular, Figure 5 reveals the class of words predicted when the preceding *sequence* of words had the form *that Noun Verb* (where the category of the current word appears at the right). As one should expect, all verbs are predicted with appreciable strength and the period is strongly predicted.

One may conclude, therefore, that the barely perceptible verb and period predictions in Fig. 4 merely reflect the fact that nouns follow verbs (during both training and testing) vastly more often than verbs/periods do. Indeed, verbs are followed by verbs in less than 3% of verb occurrences; likewise for the verb-period pattern.

Figure 6 indicates predictions made when the current input word is 'that'. Our grammar permits both nouns and verbs to appear immediately after 'that', and this accords nicely with results in Fig. 6. In the training and test corpora, relative clauses of both

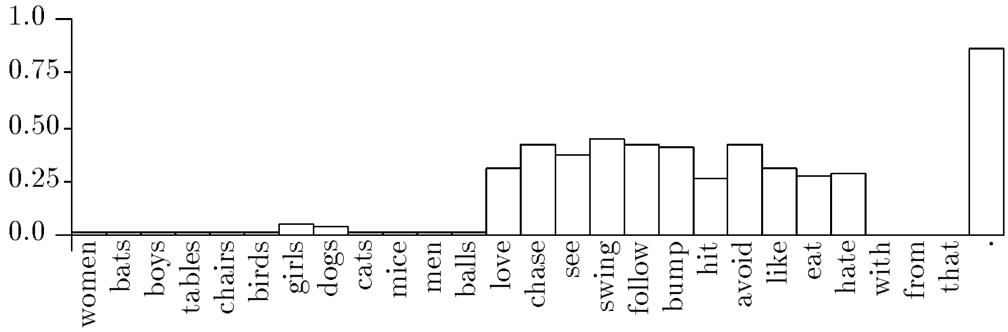

Figure 5: Predictions when the preceding sequence of words had the form: *that N V*.

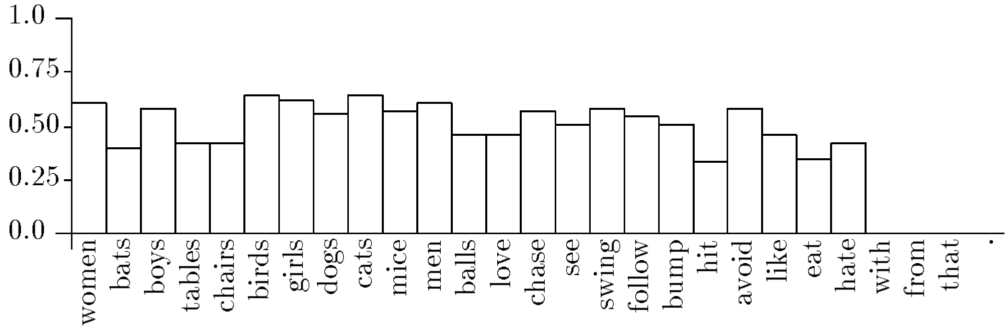

Figure 6: Predictions when the current input is 'that'.

types (*that V N* versus *that N V*) occur with about equal frequency, and this accounts for the fact that both nouns and verbs are strongly predicted to follow 'that'.

It is revealing that the "bars" for nouns and verbs in Fig. 6 are not as high as the bars in figures 4, 8, 9 and 10. This is explained by the fact that, since both nouns and verbs frequently follow 'that', the *average* vector predicted by 'that' contains substantial components for both noun and verb features. Thus, when we compare the vector for any specific noun or verb to this average vector, we find that the two vectors are not strikingly close to each other. Hence the lower cosines revealed by the bars. Comparable "lower bar" effects may be seen in Fig. 3, Fig. 5 and Figure 7, where differing word categories *frequently* follow the current input item.

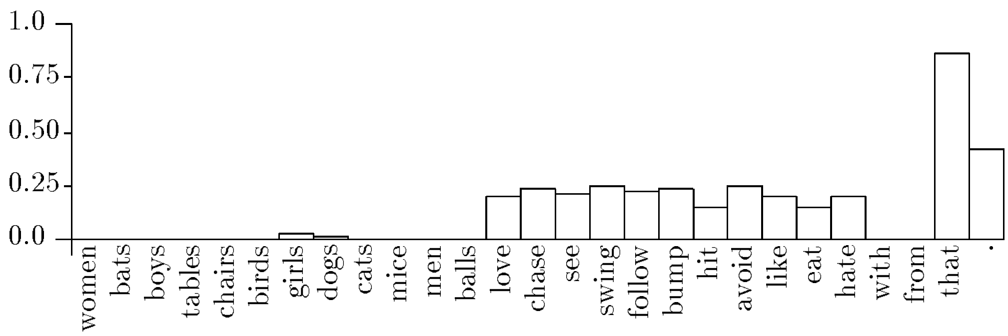

Figure 7: Predictions when the preceding sequence of words had the form: *that V N*.

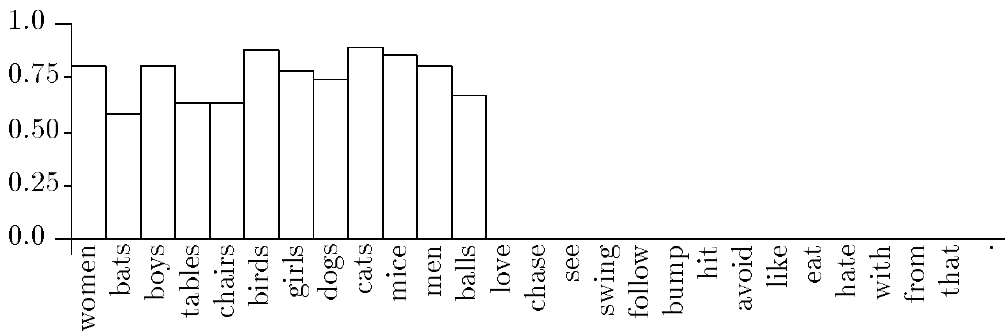

Figure 8: Predictions when the preceding sequence of words had the form: *N that V* .

Figure 7 correctly reveals the semantic categories expected when the preceding sequence of words had the form *that V N*. This particular relative clause pattern can only be followed by a verb, by a period, or by 'that' (which introduces another relative clause).

Figure 8 displays the network's expectations when the preceding sequence of words had the form *N that V*. The network's predictions accurately reflect what our grammar permits. Because nouns alone can follow the given pattern of words, we should expect bars for nouns to be strikingly high in Fig. 8. This is indeed the case.

In Figure 9 we find a similar situation. Here the network correctly predicts the semantic category to follow word sequences of the form *N that N*. In Figure 3 we saw that, generally speaking, nouns can be followed not only by verbs, but by prepositions, 'that',

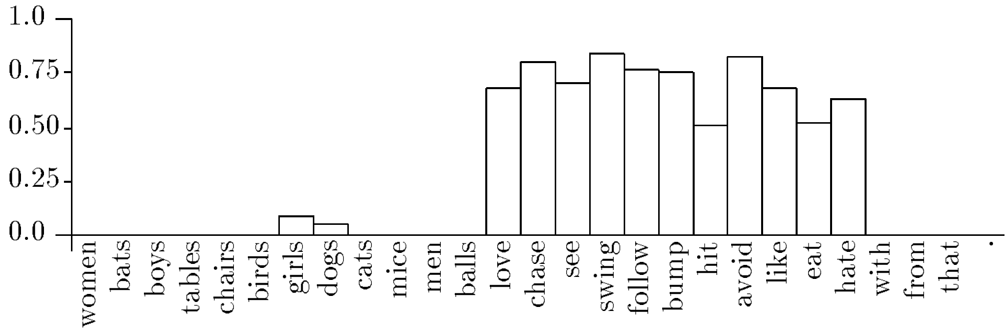

Figure 9: Predictions when the preceding sequence of words had the form: *N that N* .

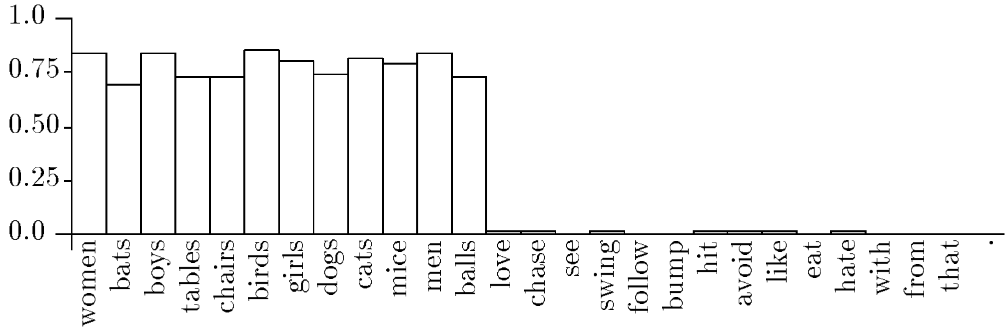

Figure 10: Predictions when the current input is a preposition.

and the period. However, when the prior word sequence is *N that N*, our grammar will permit only verbs to appear next. Within Fig. 9, only verbs have bars of significant height. The slight bars for 'dogs' and 'girls' merely reflect (as previously noted) our experimental decision to ensure that those two nouns shared one feature each with animate verbs.

Our last graph, Figure 10, correctly reveals that when the current input belongs to the category 'preposition', only nouns can be expected to follow.

In passing, it should be remembered that all cosine values reflected in the above graphs and tables were based on training sessions involving the presentation of 15,000 sentences. Significantly, however, *even after a mere 500 sentences* have been presented for training (and continuously thereafter) all cosine values are nearly at the maximums here presented (i.e, they are within at least 90% of the maximum values).

10. Discussion

In the foregoing, we presented a Hebbian-competitive network which satisfies a definition of *strong systematicity* first offered in (Hadley, 1994b). As previously noted, that definition requires that an agent *learn* to generalize the use of a *significant fraction* of its vocabulary to novel syntactic positions. It is our belief that the capacity to attain this particular form of systematicity constitutes an essential aspect of general language acquisition.

However, as previously stressed, we do not claim that our network satisfies every strong form of linguistic generalization that has been discussed in recent years. In particular, it does not satisfy generalization challenges which *presuppose* that the learning agent already possesses considerable linguistic competence and other high-level cognitive skills. We believe that certain generalization tasks posed by Marcus (1998) and Phillips (2000) fall into this latter category.

For example, Marcus presents the following as an instance where *eliminative* connectionist networks fail to *generalize outside their training space* as humans are able to do. Suppose a linguistically competent human is presented with the following series: "A rose is a rose", "A frog is a frog", "A pencil is a pencil". Humans will typically have no difficulty inducing the general pattern and completing the following sentence: "A blicket is a ...". Humans will succeed here even though 'blicket' is a novel word which is "outside their training space". Marcus uses this and other examples to argue that when humans generalize in this radical fashion, they extract symbolic rules whose application involves variable binding (Marcus, 1998; in press).

In a somewhat similar vein, Phillips (2000) presents a number of tasks which involve what he calls "cross-category" generalization. He maintains, for instance, that 'mango' belongs to a different *feature category* from colour words such as 'red', 'blue', etc. Phillips then argues that humans, but not backpropagation networks, are able to generalize sensibly from the following training data:

"John painted the chair red. What colour is the chair?" (Target answer given to the network: Red).

"John painted the chair blue. What colour is the chair?" (Target answer given to the network: Blue).

When a human agent of (at least) average intelligence is presented with the two samples just shown, and then presented with the following test case, "John painted the chair mango. What colour did John paint the chair?", the human will typically respond with 'mango'. In contrast, all artificial networks Phillips considers will fail to provide this obvious answer. Phillips argues that this failure is inevitable, provided we assume (as he thinks we ought to) that 'mango' has not been assigned semantic features which overlap with those of typical colour words. One may question whether, in fact, 'mango' possesses no significant overlap in semantic features with standard colour words, but we will not pursue that issue here. Rather, let us note that Phillips' example, like that of Marcus, presupposes considerable linguistic competence on the part of the human agents in question. Moreover, as Phillips observes, humans solve the problem by *making inferences*. Indeed, we can go farther; humans don't actually need to be *trained* on the sample set shown above. They fall back on sophisticated, pre-existing cognitive skills which exploit, among other things, their knowledge of the semantics of 'to paint'. If, out of the blue, we say to a human (Pam) "Mary painted the chair violet – what colour is the chair?", Pam can answer immediately. No **new** synaptic weight modification occurs in the time between Pam's hearing of the sentence and her responding with 'violet'. As emphasized in Hadley (1993), in cases where humans make virtually instantaneous inferences (and likewise when they acquire new rules in a matter of mere seconds), rapid synaptic weight change can be ruled out. Synapses simply do not grow anywhere near fast enough to permit coherent learning of *general* rules or general inference patterns within the span of a few seconds.

The same point applies to the Marcus example considered above. Immediately upon hearing “A rose is a rose” and “A frog is a frog”, humans are able to provide ‘blicket’ in response to “A blicket is a ...”. This ability is not a mere side effect of hearing the pattern “an X is an X” many times in the past. For precisely analogous, virtually instantaneous pattern induction occurs when we present humans with “A piffle rose transcends a zarple rose”, “A piffle frog transcends a zarple frog”, “A piffle blicket transcends a zarple ...”. In this case, the general pattern being induced is a novel one. Marcus may well be correct in concluding that the human ability to discover general patterns in cases such as these involves symbolic rule induction, where the application of such rules entails variable binding. Related and supportive conclusions were derived by Hadley in (Hadley, 1990; 1993), where it was argued that rules that are acquired virtually instantaneously, and used immediately thereafter, must be stored not as weight-embedded structures, but as active patterns in short term memory. The capacity to *apply* these activation patterns to particular data presupposes the prior acquisition of several powerful cognitive skills.

In any case, the point we are now making is that the human ability to perform very rapid pattern induction (as in the Marcus examples) and rapid, systematically related inferences (as in the Phillips case), presupposes not only a prior linguistic competence on the learner’s part, but may well involve other cognitive abilities that, for all we know, emerge only after the learning system has acquired the kind of strong systematicity attained by the network we have presented in the present work.

Turning now to other issues, a frequent complaint about connectionist models purports that the trained networks involved are black boxes; the networks may perform the required task, but it is often difficult to understand how and why. We believe this complaint is at times unfair – a deeper understanding of connectionist training processes may often unravel the mystery. In any case, we believe the success of the network presented here can be understood at a useful, relevant level of abstraction. It may be helpful at this point to explain just why our network succeeds at the systematicity task. There are two principal reasons, which are:

I. In our experiment, the assignment of a given input vector to a given semantic category, by a winning node in region A (HL1), does not depend, to any degree, on whether the input vector being processed occurs in a novel position or not. Rather it depends on whether the set of input features share some common elements with semantic vectors belonging to words previously encountered. As noted in preceding sections, given the learning rate and algorithm we employ, *all* nouns we employed were recognized by precisely the same 'winner node' in region A. This occurred *not* because all our nouns possessed very similar feature vectors. In fact, in the training-test sessions reported above, the degree of overlap in feature vectors for nouns varied between under 50% and 70%. The least overlap occurred between 'baseballs' and 'men', which shared only four of the 26 candidate features for nouns. Yet,'baseballs' and 'men' were recognized by the same winning node, during the entire test phase, as all other nouns.

Interestingly, in a separate experiment, we modified the feature set assigned to an arbitrarily chosen noun ('bats'). For this experiment, we assigned 'bats' only two positive features (features 2 and 11). Moreover, only one of these two features was shared by all nouns in common. Our network was then trained and tested using the same sentence corpora as previously described. During testing, we monitored region A winners for all words and found that 'bats' selected the very same winner within region A as every other noun. This was not surprising. For although 'bats' shared just two features with other nouns, it had no features in common with other lexical items. In general, when competitive learning is employed, a given input vector, V, will frequently be assimilated to whatever existing feature categorization most resembles V, *even though* the degree of overlap is small. For this reason among others, it is worth remembering that competitive networks have markedly different properties from the backpropagation networks Marcus and Phillips consider.

Now, admittedly, substantially altering our chosen constants (either within our *winner selection rule*, or the learning rate we have chosen), can cause *sub-categories* of noun classifications to be formed within region A. However, sub-categories would not present a problem for our systematicity results provided the sub-categories did not simply form along the lines of NS, NDO, and unrestricted nouns. We see no reason to believe that, in ordinary human language acquisition, those three classes of nouns would possess disjoint (or even largely disjoint) sets of feature vectors. It seems far more likely that, during learning, a considerable variety of semantic types of nouns occur in both subject and direct-object positions.

Now, an objection may arise that the tendency noted above, for overlapping feature vectors to be assimilated to the same "winner category", merely reinforces one of Marcus' concerns about the deployment of distributed feature representations for distinct lexical categories. Marcus maintains that separate lexical categories (e.g., nouns and verbs) can share *some* features (e.g., 'animate') and that using distributed feature vectors for lexical input items will cause a pernicious conflation of distinct lexical categories (via the overlap in features). However, we believe the results presented here should allay Marcus' worries on this issue. For, as noted above, the assimilation of overlapping input vectors to the same category occurs *when* the given vectors more closely resemble each other than they do other clusters of vectors. As exemplified in our experiments here, vectors will *not* be assimilated to a single group when (a) the degree of overlap is small *and* (b) a better match (higher degree of overlap) occurs with some competing cluster of vectors. Thus, although a small feature overlap can be found in selected noun-verb pairs in our input lexicon, it nevertheless remained true that verb input vectors never selected the "noun recognizer" in region A, and noun input vectors never selected the "verb recognizers" in region A.

II. We turn now to the second major reason why our model succeeds at the systematicity task. As noted in section 7, our learning algorithm induces a very fine-grained pattern recognition capacity within the second hidden layer, HL2. Indeed, *each distinct triplet* of activation, jointly representable by the three regions of HL1, is recognized by a unique winner in HL2. Thus, between HL1 and HL2, there is no loss of quality in distinguishing between the various distinct, but frequently overlapping "triplet contexts" (activation triples) that can be jointly recorded by the regions of HL1. Taken in conjunction with the fact that region A (of HL1) correctly classifies input vectors, without regard to their order of presentation within the input stream, winner nodes in HL2 necessarily display systematicity. Given this, a simple variant of Hebbian learning (on the topmost set of links) suffices to ensure that appropriate activation vectors within the output layer will be predicted whether or not the input stream exhibits the type of positional novelty at issue here.

Summary

In the foregoing, we have presented a Hebbian-competitive network which, in the complete absence of error feedback, acquires a strong form of systematicity. This systematicity, at a syntactic level, was shown to emerge from the reliability of predictions made for semantic features of denoting terms. In addition, the network learned to predict the appearance of syntactically key words, such as prepositions and relative pronouns.

Apart from eschewing error feedback, we imposed a number of demanding restrictions, calculated to bring some measure of cognitive plausibility to our training regime. For example, during training, *two-thirds* of all nouns were restricted to a single syntactic position. Moreover, a *relatively sparse* training corpus was employed; the network was exposed to 3000 distinct sentences, while the set of *potential* test sentences numbered well over 300 million. Also, we ensured that training and test data were generated from a non-deterministic grammar. Both subject-relative and object-relative clauses could occur within the same sentence. Relative clauses and prepositional phrases could likewise occur in the same sentence.

Despite these restrictions, during testing, the network successfully predicted the appearance of hitherto restricted nouns in all grammatically permissible contexts. The strengths of predictions for restricted nouns were virtually indistinguishable from those of unrestricted nouns. We also saw that the network successfully generalized its predictions to a deeper level of clausal embedding than ever encountered during training. Indeed, given the network's structure, tail recursion of relative clauses can be arbitrarily deep with no loss of precision in the quality of semantic predictions made. For, the contextual representations recorded by the three regions of HL1 will be identical, regardless of the depth of tail recursion.

Appendix A: Assignment of Features to Lexical Items.

number of words: 27;
number of features: 51;
number of types: 5;

		animate	inanimate	two-legs	four-legs	talks	barks	meows	squeaks	has-weight	has-size	has-shape	has-location	furry	large	small	heavy	light	laughs	bites	long-snout	flat-face	small-nose	rigid	flexible	tubular	round	physical-motion	feeling-nice	feeling-bad	emotive	rapid	slow	involves-contact	involves-animate	involves-perceiving																
women	n	1	0	1	0	1	0	0	0	1	1	1	1	0	1	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
bats	n	0	1	0	0	0	0	0	0	1	1	1	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
boys	n	1	0	1	0	1	0	0	0	1	1	1	1	0	0	1	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
tables	n	0	1	0	1	0	0	0	0	1	1	1	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
chairs	n	0	1	0	1	0	0	0	0	1	1	1	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
birds	n	1	0	1	0	0	0	0	0	1	1	1	1	0	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
girls	n	1	0	1	0	1	0	0	0	1	1	1	1	0	0	1	0	0	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
dogs	n	1	0	0	1	0	1	0	0	1	1	1	1	1	1	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cats	n	1	0	0	1	0	0	1	0	1	1	1	1	1	0	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mice	n	1	0	0	1	0	0	0	1	1	1	1	1	1	0	1	0	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
men	n	1	0	1	0	1	0	0	0	1	1	1	1	0	1	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
balls	n	0	1	0	0	0	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
love	v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
chase	v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
see	v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
swing	v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
follow	v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
bump	v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
hit	v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
avoid	v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
like	v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
eat	v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
hate	v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
with	p	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
from	p	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
that	t	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
.	.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

References

Chalmers, D. 1990, Why Fodor and Pylyshyn were wrong: the simplest refutation. *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society* (Hillsdale, NJ: Lawrence Erlbaum associates): 340-347.

Christiansen, M.H. & Chater, N., 1994, Generalization and connectionist language learning. *Mind and Language*, 9: 273-287.

Christiansen, Chater, & Seidenberg, 1999, (eds.) Special Issue on Connectionist Models of Human Language Processing: Progress and Prospects, *Cognitive Science*, 23.

Elman, J.L., 1998, Generalization, simple recurrent networks, and the emergence of structure. Paper version of a symposium talk presented to The Twentieth Annual Conference of the Cognitive Science Society, Madison, Wisconsin. (available by web at http://crl.ucsd.edu/ elman/).

Elman, J.L., 1993, Learning and development in neural networks: the importance of starting small. *Cognition*, 48: 71-99.

Elman, J.L., 1990, Finding structure in time. *Cognitive Science*, 14: 179-212.

Elman, J.L., Bates, E.A., Johnson, M.H., Karmiloff-Smith, A., Parisi, D., Plunkett, K., 1996, *Rethinking Innateness: A Connectionist Perspective on Development* (Cambridge, MA: MIT Press).

Hadley, R.F., in press, Systematicity in Connectionist Generalization. In M.A. Arbib (ed.) *The Handbook of Brain Theory and Neural Networks*, 2nd edition (Cambridge, MA: MIT Press).

Hadley, R.F., 1994a, Systematicity in connectionist language learning. *Mind and Language*, 9: 247-272.

Hadley, R.F., 1994b, Systematicity revisited: reply to Christiansen and Chater and Niklasson and van Gelder. *Mind and Language*, 9: 431-444.

Hadley, R.F., 1993, Connectionism, Explicit Rules, and Symbolic Manipulation. *Minds and Machines*, 3: 183-200.

Hadley, R.F., 1990, Connectionism, Rule Following, and Symbolic Manipulation. *Proceedings of AAAI-'90, the American Association for Artificial Intelligence* (Boston: MIT Press).

Hadley, R.F. and Cardei, V.C., 1999, Language Acquisition from Sparse Input with No Error Feedback. *Neural Networks*, 12: 217-235.

Hadley, R.F. & Hayward, M.B., 1997, Strong semantic systematicity from Hebbian connectionist learning. *Minds and Machines*, 7: 1-37.

Hebb, D.O., 1949, *The Organization of Behaviour* (New York: Wiley).

Marcus, G., 1998, Rethinking eliminative connectionism. *Cognitive Psychology*, Vol. 37.

Marcus, G., in press, *The Algebraic Mind.* (Cambridge, MA: MIT Press).

Niklasson, L.F. and van Gelder, T., 1994, On being systematically connectionist. *Mind and Language*, 9: 288-302.

O'Grady, W., 1997, *Syntactic Development* (Chicago: The University of Chicago Press).

Phillips, S., 1994, Strong systematicity within connectionism: the tensor-recurrent network. *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society* (Hillsdale, NJ: Lawrence Erlbaum associates): 723-727.

Phillips, S., 2000, Constituent similarity and systematicity: the limits of first-order connectionism. *Connection Science*, 12: 1-19.

Rispoli, M., 1999, Review essay of: Elman, J., Bates, E., Johnson, M., Karmiloff-Smith, A., Parisi, D., & Plunkett, K. Rethinking innateness: a connectionist perspective on development. *Journal of Child Language*, 26: 217-225.

von der Malsburg, C., 1973, Self-organizing of orientation cells in the striate cortex. *Kybernetik*, 14: 85-100.

Voll, K., & Hadley, R.F., in preparation, Positional Generalization in Simple Recurrent Networks.