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

Robert F. Hadley (hadley@cs.sfu.ca)
Dirk Arnold (dirk@cs.sfu.ca)
Vlad Cardei (vcardei@cs.sfu.ca)
School of Computing Science
Simon Fraser University
Burnaby,B.C., V5A 1S6, Canada

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. Moreover, the network can integrate novel nouns and verbs into its training process. This is achieved by assigning predicted semantic features as a default meaning when a novel word is encountered. All network training is unsupervised with respect to error feedback. Issues addressed here have been the subject of debate by notable psychologists, philosophers, and linguists within the last decade.

Introduction.

Between 1990 and 1995, substantial research was directed at demonstrating the capacity of simple recursive networks (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; Hadley, 1994a, 1994b; Hadley & Hayward, 1997; 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 & Hayward, 1997), 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. Moreover, we know of no connectionist network which has achieved systematicity in a *strong or robust* form except networks which employ classical (explicitly combinatorial) semantic representations within the output layer. (See Hadley & Hayward, 1997, for one such example).

This paper presents a connectionist system 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. In this context, 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.

By contrast with the SRN based architectures 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 working assumption here is that, to a large degree, systematicity at the syntactic level derives from predictability at a semantic level. Now, although we recognize that certain of our strategies could just as well be implemented in some version of SRN, trained by the standard backpropagation algorithm, we have sought to avoid all forms of error feedback. Several researchers have remarked upon the desirability of replacing backpropagation-based networks with architectures which are (at least) closer to biologically grounded systems. For this reason, we have employed only Hebbian and self-organizing forms of connectionist learning (see Hebb, 1949; Rumelhart and Zipser, 1986).

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. During training, the network is exposed to less than 4000 distinct sentences, while the set of *potential* test sentences numbers over 300 million. Also, during training, two-thirds of all nouns are restricted to a single syntactic position and clausal embedding is restricted to a depth of one. During testing, these restrictions are dropped. Moreover, once training is underway, the network is presented with *novel* nouns and verbs. The network's learning is not derailed by exposure to these novel items; rather default semantic features are assigned to these words and learning progresses unhindered.

Task and System 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 nouns and verbs shown in Fig. 1 have previously been assigned semantic feature vectors. The totality of possible semantic features for nouns and verbs are 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. In addition to the basic vocabulary shown in Fig. 1, three of four distinct training sets (corpora) contain dozens of sentences in which novel nouns and verbs

S → NP V NP
NP → N | N RC | N PP
N → women | girls | birds | bats | men | boys | chairs | balls | dogs | tables | cats | mice
V → chase | sees | swing | love | avoid | follow | bump | hit | consume | dislike
RC → that V NP
PP → Prep NP
Prep → from | with

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

appear. Similar to other training data, these sentences follow the syntax given in Fig. 1. The novel words in question have no previously assigned semantic features.

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 (*next*) in a sentence, when given semantic features for the current word (*current*). However, if features are not available for *next* at the time *current* is presented, an attempt is soon made to assign reasonable semantic features to *next*. In certain cases, where sensible semantic features cannot be discovered, the network can learn to predict the probable occurrence of certain words. For this reason, the network's output layer contains not only a semantic region, but a lexical region.

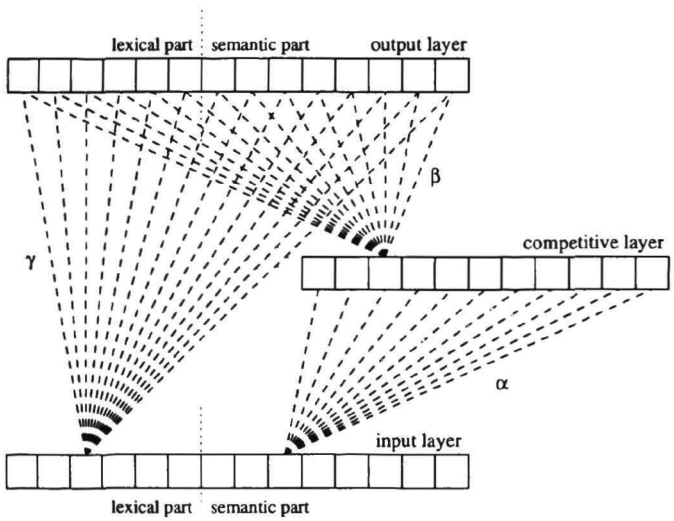

Figure 2: Overall Architecture of the Hebbian-Competitive network.

In order to challenge the generalization capacity of our network, the various training corpora are designed to ensure that 2/3 of all nouns are limited to a single syntactic position *during training*. Each sentence selected from a given training corpus is presented, one word at a time, to the input layer of the network. As a word is provided as input, its lexical encoding is activated within the lexical region of the input layer (see Fig. 2). In addition, if the word's semantic features are already known, they are activated within the input layer's semantic region.

To enable the system to learn to predict *typical* semantic features, any available 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.

Training Corpora.

There are four separate training corpora; each containing 1000 sentences. Each sentence in each corpus employs the syntax displayed in Fig. 1. Corpus 1 contains only the vocabulary presented in the original grammar. The remaining three corpora each include some sentences involving novel nouns and verbs whose semantic features are initially unknown. Corpus 2 contains just one novel noun and a novel verb in addition to all previously employed vocabulary. An additional two nouns and two verbs are added when corpus 3 is generated; similarly for corpus 4. Thus, a total of ten novel nouns and verbs are introduced.

In all four corpora, about 50% of sentences are of the simplest form (noun verb noun). The remaining sentences all contain either one relative clause or one prepositional phrase. For purposes of testing *syntactic* systematicity, four of the initial nouns were permitted to appear only as grammatical *subject*, and another four could appear only in *object* position. The remaining four initial nouns were not restricted. However, of the initial set, only these 'free' nouns were permitted to appear immediately before 'that', which introduces the relative clauses.

In addition to the above, some semantic constraints were employed. Only one of the initial 'inanimate nouns' is ever employed as grammatical subject during training, and this occurs only in conjunction with the verbs 'bump' and 'hit'. Every training sentence is generated in a fashion which maximizes randomness. That is, subject to all constraints mentioned above, whenever a decision is made about which word to pick, or whether to employ a simple or complex sentence, a random selection is made.

Algorithms and Architectural Details.

As indicated in Fig. 2, there are lexical and semantic regions within both the input and output layers. For convenience, we refer to these as 'lexical-in', 'semantic-in', 'lexical-out' and 'semantic-out'. Both lexical-in and lexical-out employ local encodings, where a single unit (one of 36) is assigned to represent a single word. The end of sentence marker (a period) is also assigned a single unit within the lexical layers. Likewise, the representation of a single semantic feature is local, but the representation of an entire feature vector, corresponding to a word's meaning, is distributed across several units within both semantic-in and semantic-out. Since there are a total of 35 semantic features, both semantic-in and semantic-out contain 35 units.

The only hidden layer present (middle of Fig. 2) should be regarded as a single competitive cluster. It contains 20 units which compete with one another to represent patterns of semantic input values. All input

units and competitive units produce only binary (1,0) output. By contrast, units within the top output layer can produce real values as output.

As indicated in Fig. 2, three sets of links (α, β, and γ) connect various regions/layers. Each of the three sets comprises a *fully connected* link configuration (all pairwise combinations of nodes selected from the adjoining layers are linked). Weights on the α, β, and γ sets are initialized as follows: α-links, which connect semantic-in with the competitive layer, are initialized with random values from the interval $[0, 1]$ and are then normalized such that for each unit j in the competitive layer, the sum of weights of all links feeding into the unit equals 1. All other weights are initialized to 0.

Training Algorithms.

The α links, which connect semantic-in with the competitive layer, are trained by means of a familiar Hebbian-inspired, competitive learning algorithm, developed by von der Malsburg (1973). (See Rumelhart & Zipser, p. 164, for a concise explanation). The network's learning rate is set to .01. However, we do employ one variation on the typical use of this algorithm. In most applications (though not all) a single winner is selected on a given iteration from a competitive cluster. By contrast, we select, on each training iteration, the five most active nodes as winners and update weights on links feeding into each of these nodes. Once a given winner is selected, it fires and sends an output of +1 towards the output layer. Competitive "losers" transmit no output.

By contrast with α links, β and γ links do not enter a hidden layer, and are trained according to a simple Hebbian-based formula. On each iteration, an increment is applied to β and γ links, provided some positive activation has just passed through that link to an output node that was already active. Thus, increments are applied only in cases where both the sending and receiving nodes begin with positive activation levels. Each increment is computed thus:

$$\text{increment} = 1.25 \cdot 10^{-5}$$

The Training Cycle.

As previously mentioned, there are four distinct corpora. Sentences from a given corpus are randomly selected for training. On average, each sentence from each corpus is presented to the network 12 times. For each selected sentence, the following cycle is applied:

Let *current* be the first word in the sentence.

REPEAT until current = '.'

Let *next* = the item following *current*.
Activate *current's* lexical encoding within lexical-in.
If *current* has an available semantic feature
  vector, activate that within semantic-in;
  Else, semantic-in remains inactive and
    transmits no activation
    on this iteration.
If *next* has an available semantic encoding,
  activate that within semantic-out;
  Else, if *next* has previously been
    flagged as "meaning unknown"
    (it still lacks a semantic encoding),
    Then activate the lexical encoding
    of *next* within lexical-out.
    Else, place *next* on Hold and place
      zero activation in semantic-out.
Spread activation upwards from all
  active input units.
Apply *competitive* and Hebbian training
  to α and γ links
  (respectively) wherever possible.
If activation did spread from semantic-in
  to the competitive layer then some
  winners emerged. Let those winners fire
  along β links. Apply Hebbian training,
  wherever appropriate, to β links.
If *next* was put on Hold, then examine
  current activations reaching
  units in semantic-out.
  If the activation level of any such unit
  is at least 20% of the maximum activation
  thus far received by that unit,
  then a reasonable semantic feature was
  predicted for the word in *next*.
  So, assign each such semantic feature as
  a default feature for that word.
  Default semantic features for a given
  word are recorded for later use.
    Else, no reasonable semantic prediction
    was made for the word in *next*.
    So, flag that word as "meaning unknown".
Let *current* be assigned the contents of *next*.
END OF CYCLE.

As a careful reading of the above outline reveals, the process governing precisely *how* default semantic features are assigned to words is implemented, in some aspects, by a procedure external to the network itself. (The same can be said of many other connectionist systems; consider, for example, the "probing" process used in St. John's and McClelland's (1990) approach to language learning.) However, the default semantic features which are actually assigned to novel words are those which the network actually predicts. Moreover, we see no reason why the "external processes" could not arise through the activity of external *connectionist* modules.

On a separate theme, it is noteworthy that novel words *can* be assigned default semantic content if training has progressed sufficiently. However, in our training corpora, the words 'with', 'from' and 'that' are encountered too early for reasonable semantic content to be assigned. In such cases, the algorithm treats the words merely as lexical items. This decision is not reversed unless the words are later assigned semantic features by external means. Such external means might correspond to ostensive definition or verbal explanations.

Although such "early occurring" words are never assigned default semantic content by the network, they still play a predictive role. For example, as our test results reveal, 'that' consistently predicts the occurrence of verbs, and prepositions overwhelmingly predict the occurrence of nouns. Moreover, because the lexical encodings of such words are frequently activated within lexical-out, the network learns to predict their occurrence in a reliable fashion. So, although the network's primary task is to predict semantic content, it also learns to predict the occurrence of particular words whose function seems more syntactic than semantic.

The Test Corpus.

A single 3000 sentence test corpus was generated using the grammar previously discussed. In addition to the ba-

sic vocabulary (Fig. 1), all novel nouns and verbs were permitted to appear within sentences. Apart from semantic sensibility constraints, previous restrictions were relaxed. Every noun could freely serve as subject or object, and as head of a relative clause. Within the test corpus, ten percent of all sentences involved deep embedding of relative clauses, to a maximum depth of three levels. An additional fifteen percent of test sentences contained either a relative clause or a prepositional phrase. As with the training corpora, random selections were made wherever possible.

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 lexical encoding was activated within lexical-in and, if available, its semantic features were activated within semantic-in. Once activation had propagated to the output regions, activation vectors appearing in lexical-out and semantic-out were accumulated for later averaging. Various cosine comparisons were made both between particular average vectors and also between average vectors and the semantic vectors of particular words. These comparisons are discussed below and are displayed in Figures 3, 4, 5, 6, and 7.

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. In addition, activations for the words 'with', 'from', and 'that' are accurately predicted within lexical-out. Semantic features for these words are *not* predicted because the network never assigns features to these words.

Significantly, abundant evidence for syntactic systematicity did emerge 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. The cosine of the angle between the two average vectors was above .99, indicating that the vectors were very close to each other. The same kind of comparison was performed for those nouns whose positions were unrestricted during training; the result was virtually identical. In this case, the cosine was above .993. Precisely similar analyses were performed for output vectors in lexical-out. The results here were even better. The cosine for restricted nouns was nearly .9986 and for unrestricted nouns, the result was above .999. Clearly, the predictive power of both restricted and unrestricted nouns is very good and nearly equal. Included in all these analyses are those "novel" nouns whose semantic vectors were assigned by the network itself during training.

Our remaining analyses were intended to confirm that words belonging to *each* syntactic category (noun, verb, preposition, and relative pronoun) made strong predictions for semantic features, or lexical representations, belonging to words which could legally follow the word making the prediction. For example, Figure 3 indicates which words possess semantic features that are strongly predicted when the input word belongs to the category 'noun'. In detail, the height of each bar in the graph

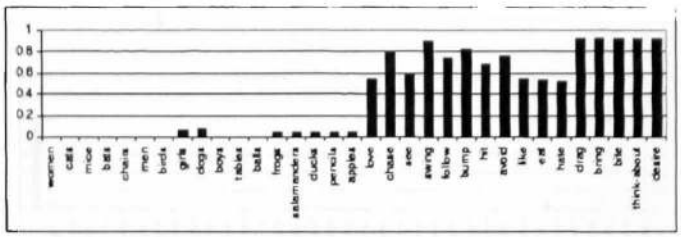

Figure 3: Semantic predictions when the input is a noun.

displays, in terms of cosines, how closely the semantic feature vector of each vocabulary word matches the *average* vector produced within semantic-out when the input word is a noun. As can be seen, only the class of verbs possesses semantic features close to those predicted by the class of nouns. Interestingly, the five verbs at the right side of the graph, whose vectors are closest to the average predicted vector, are 'novel' verbs, which were assigned default features during the training phase. The strength of these predictions is explained by the fact that default features are the very features that are predicted by an input word on a given occasion. In the present case, all the assigned default features were predicted when the input was a noun. Note also that certain very weak predictions for specific nouns can be discerned. We believe these weak predictions are due to randomness that arises in the early stages of the learning process.

Although verbs are the only words whose *semantic features* are strongly predicted by nouns, other words can, of course, grammatically follow a nouns. Within our restricted language, either of two prepositions ('with' and 'from') or one relative pronoun ('that') can legally fill this position. However, these words were assumed *not* to have known semantic features at the onset of training, and they never acquired features during the course of training (due to the fact that they are each first encountered before any semantic predictions of sufficient strength are made by the network). The upshot is that semantic features for these words are not predicted within semantic-out. Nevertheless, as revealed in Figure 4, the network does learn to predict the appearance of each of these three words following the occurrence of a noun. Figure 4 displays *activation levels* predicted for units within lexical-out when the input item is a noun. (Cosine measures would have been inappropriate in this analysis because each word's lexical encoding contains only a single "on" bit.) The displayed activation levels reflect the raw frequency of occurrence of particular words. The very strong prediction for the period arises from the fact that, within our corpora, nouns are followed by a period much more often than by any of the three words in question.

Figure 5 illustrates the strength of *semantic* predictions for semantically known words when the input belongs to the category, 'verb'. As with Fig. 3, the height of each bar reflects the cosine between each word's known feature vector and the average semantic vector predicted when the input word is a verb. Clearly, all and only the semantic features of nouns are predicted with any significant strength.

Figures 6 and 7 are precisely analogous to the preceding graph. In Figure 6 we see that only semantic features

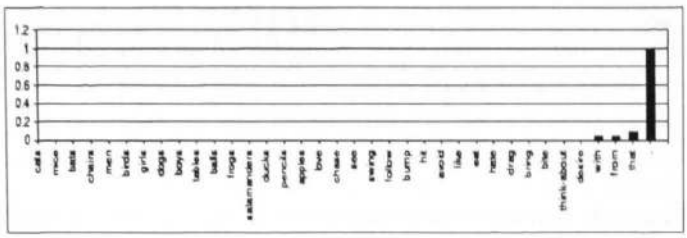

Figure 4: Lexical predictions when the input is a noun.

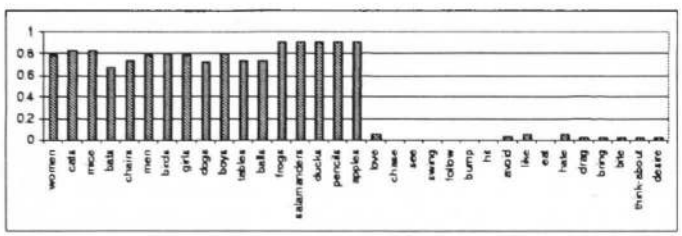

Figure 5: Semantic predictions when the input is a verb.

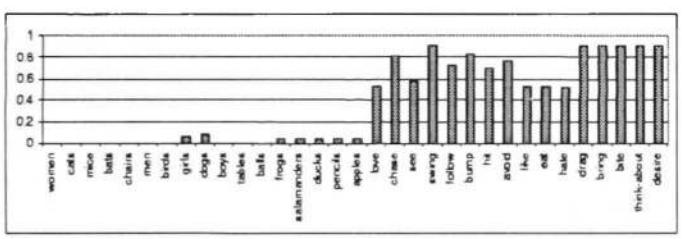

Figure 6: Semantic predictions when the input is 'that'.

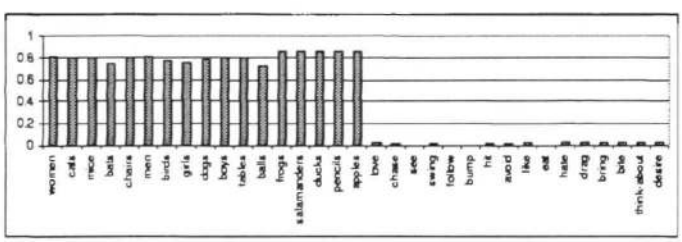

Figure 7: Semantic predictions when the input is a preposition.

for verbs are strongly predicted when the input is a relative pronoun ('that'). This is just what we would hope, given the grammar in Fig. 1. Likewise, Figure 7 reveals that only noun features are strongly activated when the input word is either 'with' or 'from'.

Summary and Future Directions.

As the preceding analysis implies, the stated goals for our working model have been attained. Clearly, the network does learn to make accurate semantic predictions, when provided with words taken from any of the syntactic categories listed in the target grammar. In addition, the level of accuracy was virtually identical both for nouns whose position was restricted during training, and for unrestricted nouns. Thus, strong systematicity in a syntactic dimension was achieved. This is despite the fact that two-thirds of all nouns were presented in a single syntactic role during training. Moreover, the network successfully integrated novel nouns and verbs into its training process. Indeed, 10 of 32 nouns/verbs used in the final test corpus were not present in the first training corpus. Significant also is the fact that all network training was unsupervised with respect to error feedback.

Finally, it is clear that the presence of deeply embedded relative clauses within the test corpus did not degrade the network's predictive accuracy. This was to be expected, of course, since the network's predictions are a function only of the current input state – no memory or context layers are contained in the network. The absence of context layers does present a limitation, however. Unlike Elman's SRN, the present incarnation of our network cannot detect long range dependencies between predicted categories. Nevertheless, by the time this paper appears, we will have implemented an extended version of the current model, which we expect to overcome this limitation. Indeed, prior, closely related research (by Cardei and Hadley, 1996) strongly suggests that context-sensitive behavior of the required kind can be achieved through the inclusion of (a) additional "memory layers", which retain prior contents of the competitive layer, and (b) the addition of a higher-level self-organizing layer which receives input from the first competitive layer and all memory layers. We wish to stress, moreover, that our present results have been attained via training algorithms which are widely believed to be closer to biological reality than the commonly used backpropagation method.

Appendix A.

Features assigned to words in our implementation are admittedly incomplete and approximate. However, they serve to convey the general approach we have adopted.

Features assigned to 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.

Note that all nouns would have certain features in common, e.g., has-weight, has-size, has-shape, has-location. Features assigned to verbs are taken to be subsets of the following:

rapid, slow, emotive, feeling-nice, physical-motion, involves-contact, involves-animate, feeling-bad, involves-perceiving.

Acknowledgements.

We wish to thank Michael Spector for his assistance in generating the training and test data sets. In addition, we are grateful to Christina Carrick for her help in resizing the graphs.

References.

Cardei, V. and Hadley, R.F. (1996). Microfeature prediction using neural networks, Technical Report available at: http://fas.sfu.ca/cs/people/GradStudents/vcardei/personal/Projects/

Chalmers, D. (1990). Why Fodor and Pylyshyn were wrong: the simplest refutation. *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, Cambridge, Mass.

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

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

Elman, J.L. (1993). Learning and Development in Neural Networks: The Importance of Starting Small. *Cognition*, 48, 71-99.

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. and Hayward, M.B. (1997). Strong semantic systematicity from Hebbian connectionist learning, *Minds and Machines*, Vol. 7, 1-37

Hebb, D.O. (1949). *The organization of behaviour.* New York: Wiley.

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

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

Phillips, S. (1994). Strong systematicity within connectionism: the tensor-recurrent network. *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, Atlanta, GA.

Rumelhart, D.E., and Zipser, D. (1986). Feature Discovery by Competitive Learning, in D.E. Rumelhart, J.L McClelland and the PDP Research Group (Eds.), *Parallel Distributed Processing; Explorations in the Microstructure of Cognition, Volume 1*, Cambridge, MA: MIT Press.

St. John, M.F. and McClelland, J.L. (1990) "Learning and Applying Contextual Constraints in Sentence Comprehension", *Artificial Intelligence*, Vol. 46, 217-257.