(In press , *Machine Learning)*

Distributed representations, simple recurrent networks, and grammatical structure

Jeffrey L. Elman
Departments of Cognitive Science and Linguistics
University of California, San Diego.

INTRODUCTION

In recent years there has been considerable progress in developing connectionist models of language. This work has demonstrated the ability of network models to account for a variety of phenomena in phonology (e.g., Gasser & Lee, 1990; Hare, 1990; Touretzky, 1989; Touretzky & Wheeler, 1989), morphology (e.g., Hare, Corina, Cottrell, 1989; MacWhinney et al, 1989; Plunkett & Marchman, 1989; Rumelhart & McClelland, 1986b; Ryder, 1989), spoken word recognition (McClelland & Elman, 1986), written word recognition (Rumelhart & McClelland, 1986; Seidenberg & McClelland, 1989), speech production (Dell, 1986; Stemberger, 1985), and role assignment (Kawamoto & McClelland, 1986; Miikkulainen & Dyer, 1989a; St. John & McClelland, 1989). It is clear that connectionist networks have many properties which make them attractive for language processing.

At the same time, there remain significant shortcomings to current work. This is hardly surprising: natural language is a very difficult domain. It poses difficult challenges for any paradigm. These challenges should be seen a positive light. They test the power of the framework and can also motivate the development of new connectionist approaches.

In this paper I would like to focus on what I see as three of the principal challenges to a successful connectionist account of language. They are:

(1) What is the nature of the linguistic representations?

(2) How can complex structural relationships such as constituency be represented?

(3) How can the apparently open-ended nature of language be accommodated by a fixed-resource system?

Interestingly, these problems are closely intertwined, and all have to do with representation.

One approach which addresses the first two problems is to use localist representations. In localist networks, nodes are assigned discrete interpretations. In such models (e.g. Kawamoto & McClelland, 1986; St. John & McClelland, 1988) nodes may represent grammatical roles (e.g., agent, theme, modifier) or relations (e.g., subject, daughter-of). These may be then bound to other nodes which represent the word-tokens which instantiate them either by spatial assignment (Kawamoto & McClelland, 1986; Miikkulainen & Dyer, 1989b), concurrent activation (St. John & McClelland, 1989), or various other techniques (e.g., Smolensky, in press).

Although the localist approach has many attractions, it has a number of important drawbacks as well.

First, the localist dictum, “one node/one concept”, when taken together with the fact that networks typically have fixed resources, seems to be at variance with the open-ended nature of language. If nodes are pre-allocated to defined roles such as subject or agent, then in order to process sentences with multiple subjects or agents (as is the case with complex sentences) there must be the appropriate number and type of nodes. But how is one to know just which types will be needed, or how many to provide? The situation becomes even more troublesome if one is interested in discourse phenomena. Generative theories of language (Chomsky, 1965), have made much of the unbounded generativity of natural language; it has been pointed out (Rumelhart & McClelland, 1986a) that in reality, language productions in practice are in fact of finite length and number. Still, even if one accepts these the practical limitations, it is noteworthy that they are soft (or context-sensitive), rather than hard (or absolute) in the way that the localist approach would predict.(For instance, consider the difficulty of understanding "the cat the dog the mouse saw chased ran away" compared with, "the planet the astronomer the university hired saw exploded". Clearly, semantic and pragmatic considerations can facilitate parsing structures which are otherwise hard to process (see also Labov, 1973; Reich & Dell, 1977; Schlesinger, 1968; Stolz, 1967, for experimental demonstrations of this point). Thus, although one might anticipate the most commonly occurring structural relations one would like the limits on processing to be soft rather than hard, in the way the localist approach would be.

A second shortcoming to the use of localist representations is that they often underestimate the actual richness of linguistic structure. Even the basic notion “word”, which one might assume to be a straightforward linguistic primitive, turns out to be more difficult to define than one might have thought. There are dramatic differences in terms of what counts as a word across languages; and even within English, there are morphological and syntactic processes which yield entities which are word-like in some but not all respects (e.g., apple pie, man-in-the-street, man for all seasons). In fact, much of linguistic theory is today concerned with the nature and role of representation, with less focus on the nature of operations.

Thus, while the localist approach has certain positive aspects, it has definite shortcomings as well. It provides no good solution to the problem of how to account for the open-ended nature of language, and the commitment to discrete and well-defined representations may make it difficult to capture the richness and high dimensionality required for language representations.

Another major approach involves the use of distributed representations (Hinton, 1988; Hinton, McClelland, & Rumelhart, 1986; van Gelder, in press), together with a learning algorithm, in order to infer the linguistic representations. Models which have used the localist approach have typically made an *a priori* commitment to linguistic representations (such as agent, patient, etc.); networks are then explicitly trained to identify these representations in the input by activating nodes which correspond to them. This presupposes that the target representations are theoretically valid; it also begs the question of where (in the real world) the corresponding teaching information might come from. In the alternative approach, tasks must be devised in which the abstract linguistic representations do not play an explicit role. The model's inputs and output targets are limited to variables which are directly observable in the environment. This is a more naturalistic approach in the sense that the model learns to use surface linguistic forms for communicative purposes rather than to do linguistic analysis. Whatever linguistic analysis is done (and whatever representations are developed) is internal to the network and is in the service of a task. The value of this approach is that it need not depend on pre-existing preconceptions about what the abstract linguistic representations are. Instead, the connectionist model can be seen as a mechanism for gaining new theoretical insight. Thus, this approach offers a potentially more satisfying answer to the first question, What are the nature of linguistic representations?

There is a second advantage to this approach. Because the abstract representations are formed at the hidden layer, they also tend to be distributed across the high-dimensional (and continuous) space which is described by analog hidden unit activation vectors. This means there is a larger and much finer-grained representational space to work with than is usually possible with localist representations. This space is not infinite, but for practical purposes it may be very, very large. And so this approach may also provide a better response to the third question, How can the apparently open-ended nature of language be accommodated by a fixed-resource system?

But all is not rosy. We are still left with the second question: How to represent complex structural relationships such as constituency. Distributed representations are far more complex and difficult to understand than localist representations. There has been some tendency to feel that their murkiness is intractable and that "distributed" entails "unanalyzable." Although, in fact, there exist various techniques for analyzing distributed representations (including cluster analysis, Elman, 1990; Hinton, 1988; Sejnowski & Rosenberg, 1987; Servan-Schreiber, Cleeremans, & McClelland, in press; direct inspection, Pollack, 1988; principal component & phase state analysis, Elman, 1989; and contribution analysis,Sanger, 1989), the results of such studies have been limited. These analyses have demonstrated that distributed representations may posses internal structure which can encode relationships such as kinship (Hinton, 1987) or lexical category structure (Elman, 1990). But such relationships are static. Thus, for instance, in Elman (1990) a network was trained to predict the order of words in sentences. The network learned to represent words by categorizing them as nouns or verbs, with further subcategorization of nouns as animate/inanimate, human/non-human, etc. These representations were developed by the network and were not explicitly taught.

While lexical categories are surely important for language processing, it is easy to think of other sorts of categorization which seem to have a different nature. Consider the following sentences.

(1) a. The boy broke the window.
b. The rock broke the window.
c. The window broke.

The underlined words in all the sentences are nouns, and their representations should reflect this. Nounhood is a category property which belongs inalienably to these words, and is true of them regardless of where they appear (as nouns; derivational processes may result in nouns being used as verbs, and *viceversa*). At a different level of description, the underlined words are also similar in that they are categorizable as the subjects of their sentences. This property, however, is context-dependent. The word "window" is a subject only in sentence (1c). In the other two sentences it is an object. At still another level of description, the three underlined words differ. In (1a) the subject is also the agent of the event; in (1b) the subject is the instrument; and in (1c) the subject is the patient (or theme) of the sentence. This too is a context-dependent property.

These examples are simple demonstrations of the effect of grammatical structure; that is, structure which is manifest at the level of utterance. In addition to their context-free categorization, words inherit properties by virtue of their linguistic environment. Although distributed representations seem potentially able to respond to the first and last of the problems posed at the outset, it is not clear how they address the question, How can complex structural relationships such as constituency be represented? As Fodor & Pylyshyn (1988) have phrased it,

> You need two degrees of freedom to specify the thoughts that an intentional system is entertaining at a time; one parameter (active vs inactive) picks out the nodes that express concepts that the system has in mind; the other (in construction vs not) determines how the concepts that the system has in mind are distributed in the propostions that it entertains. (pp. 25-26)

At this point, it is worth reminding ourselves of the ways in which complex structural relationships are dealt with in symbolic systems. Context-free properties are typically represented with abstract symbols such as S, NP, V, etc. Context-sensitive properties are dealt with in various ways. Some theories (e.g., Generalized Phrase Structure

Grammar) designate the context in an explicit manner through so-called "slash-categories". Other approaches use additional category labels (e.g., Cognitive Grammar, Relational Grammar, Government & Binding) to designate elements as subject, theme, argument, trajectory, path, etc. In addition, theories may make use of trees, bracketing, co-indexing, spatial organization, tiers, arcs, circles, and diacritics in order to convey more complex relationships and mappings. Processing or implementation versions exist for some of these theories; nearly all require a working buffer or stack in order to account for the apparently recursive nature of utterances. All in all, a rather formidable armamentarium is required.

Returning to the three questions posed at the outset, although distributed representations have characteristics which plausibly may address the need for representational richness, flexibility, and may provide soft (rather than hard) limits on processing; but we now must ask whether such an approach can capture structural relationships of the sort required for language. That is the question which motivated the work to be reported here.

There is preliminary evidence which is encouraging in this regard. Hinton (1988) has described a scheme which involve "reduced descriptions" of complex structures, and which represent part-whole hierarchies. Pollack (1988, in press) has developed a training regimen called Recursive Auto-Associative Memory (RAAM) which appears to have compositional properties and which supports structure-sensitive operations (see also Chalmers, 1989). As discussed earlier, Elman's (1990) use of Simple Recurrent Networks (SRN; Servan-Schreiber, Cleeremans, & McClelland, in press) provides yet another approach for encoding structural relationships in a distributed form.

The work described here extends this latter approach. An SRN was taught a task involving stimuli in which there were underlying hierarchical (and recursive) relationships. This structure was abstract in the sense that it was implicit in the stimuli, and the goal was to see if the network could (a) infer this abstract structure; and (b) represent the compositional relationships in such a manner as to support structure-sensitive operations.

The remainder of this paper is organized as follows. First, the network architecture will be briefly introduced. Second, the stimulus set and task will be presented, and the properties of the task which make it particularly relevant for the question at hand will be described. Next, the results of the simulation will be presented. In the final discussion, differences and similarities between this approach and more traditional symbolic approaches to language processing will be discussed.

Network Architecture

Time is an important element in language, and so the question of how to represent serially ordered inputs is crucial. Various proposal have been advanced (for reviews, see Elman, 1990; Mozer, 1988). The approach taken here involves treating the network as a simple dynamical system in which previous states are made available as an additional input (Jordan, 1986). In Jordan's work, the network state at anyh point in time was a function of the input on the current time step, plus the state of the output units on the previous time step. In the work here, the network's state depends on current input, plus its own internal state (represented by the hidden units) on the previous cycle. Because the hidden units are not taught to assume specific values, this means that they can develop representations, in the course of learning a task, which encode the temporal structure of the task. In other words, the hidden units learn to become a kind of memory which is very task-specific.

— Insert Figure 1 about here —

The type of network used in the current work is shown in Figure 1. This network has the typical connections from **input units** to **hidden units**, and from hidden units to **output units**. (Additional hidden layers between input and main hidden, and between main hidden and output, may be used to serve as transducers which compress the input and output vectors.) There are an additional set of units, called **context units**, which provide for limited recurrence (and so this may be called a **simple recurrent network**). These context units are activated on a one-for-one basis by the hidden units, with a fixed weight of 1.0, and have linear activation functions.

The result is that at each time cycle the hidden unit activations are copied into the context units; on the next time cycle, the context combines with the new input to activate the hidden units. The hidden units therefore take on the job of mapping new inputs and prior states to the output. Because they themselves constitute the prior state, they must develop representations which facilitate this input/output mapping. The simple recurrent network has been studied in a number of tasks (Elman, 1990; Gasser, 1989; Hare, Corina, & Cottrell, 1988; Servan-Schreiber, Cleeremans, & McClelland, in press).

Task and Stimuli

The prediction task. In Elman (1990) a network similar to that in Figure 1 was trained to predict the order of words in simple (2- and 3-word) sentences. At each point in time, a word was presented to the network. The network's target output was simply the next word in sequence. The lexical items (inputs and outputs) were represented in a localist form using basis vectors; i.e., each word was randomly assigned a vector in which a single bit was turned on. Lexical items were thus orthogonal to one another, and the form of each item did not encode any information about the item's category membership. The prediction was made on the basis of the current input word, together with the prior hidden unit state (saved in the context units).

This task was chosen for several reasons. First, the task meets the desideratum that the inputs and target outputs be limited to observables in the environment. The network's inputs and outputs are immediately available and require minimal a priori theoretical analysis (lexical items are orthogonal and arbitrarily assigned). The role of an external teacher is minimized, since the target outputs are supplied by the environment at the next moment in time. The task involves what might be called "self-supervised learning."

Second, although language processing obviously involves a great deal more than prediction, prediction does seem to play a role in processing. Listeners can indeed predict (Grosjean, 1980), and sequences of words which violate expectations—i.e., which are unpredictable—result in distinctive electrical activity in the brain (Kutas, 1988; Kutas & Hillyard, 1980; Tanenhaus et al, in press).

Third, if we accept that prediction or anticipation plays a role in language learning, then this provides a partial solution to what has been called Baker's paradox (Baker, 1979; Pinker, 1989). The paradox is that children apparently do not receive (or ignore, when they do) negative evidence in the process of language learning. Given their frequent tendency initially to over-generalize from positive data, it is not clear how children are able to retract the faulty over-generalizations (Gold, 1967). However, if we suppose that children make covert predictions about the speech they will hear from others, then failed predictions constitute an indirect source of negative evidence which could be used to refine and retract the scope of generalization.

Fourth, the task requires that the network discover the regularities which underlie the temporal order of the words in the sentences. In the simulation reported in Elman (1990) these regularities resulted in the network's constructing internal representations of inputs which marked words for form class (noun/verb) as well as lexico-semantic characteristics (animate/inanimate, human/animal, large/small, etc.)

The results of that simulation, however, bore more on the representation of lexical category structure, and the relevance to grammatical structure is unclear. Only monoclausal sentences were used, all all shared the same basic structure. Thus the question remains open whether the internal representations that can be learned in such an architecture are able to encode the hierarchical relationships which are necessary to mark constituent structure.

Stimuli. The stimuli in this simulation were sequences of words which were formed into sentences. In addition to monoclausal sentences, there were a large number of complex multi-clausal sentences.

Sentences were formed from a lexicon of 23 items. These included 8 nouns, 12 verbs, the relative pronoun **who**, and an end-of-sentence indicator (a period). Each item was represented by a randomly assigned 26-bit vector in which a single bit was set to 1 (3 bits were reserved for another purpose). A phrase structure grammar, shown in Table 1, was used to generate sentences. The resulting sentences possessed certain important properties. These include the following.

— Insert Table 1 about here —

(a) *Agreement*

Subject nouns agree with their verbs. Thus, for example , (2a) is grammatical but not (2b). (The training corpus consisted of positive examples only; starred examples below did not actually occur).

(2a) John feeds dogs.
(2b) *Boys sees Mary.

Words are not marked for number (singular/plural), form class (verb/noun, etc.), or grammatical role (subject/object, etc.). The network must learn first that there are items which function as what we would call nouns, verbs, etc.; then it must learn which items are examples of singular and plural; and then it must learn which nouns are subjects and which are objects (since agreement only holds between subject nouns and their verbs).

(b) *Verb argument structure*

Verbs fall into three classes: those that require direct objects, those that permit an optional direct object, and those that preclude direct objects. As a result, sentences (3a-d) are grammatical, whereas sentences (3e, 3f) are ungrammatical.

(3a) Girls feed dogs. (*D.O. required*)
(3b) Girls see boys. (*D.O. optional*)
(3c) Girls see. (*D.O. optional*)
(3d) Girls live. (*D.O. precluded*)
(3e) *Girls feed.
(3f) *Girls live dogs.

Because all words are represented with orthogonal vectors, the type of verb is not overtly marked in the input and so the class membership needs to be inferred at the same time as the cooccurrence facts are learned.

(c) *Interactions with relative clauses*

The agreement and the verb argument facts become more complicated in relative clauses. Although direct objects normally follow the verb in simple sentences, some relative clauses have the subordinate clause direct object as the head of the clause. In these cases, the network must recognize that there is a gap following the subordinate clause verb (because the direct object role has already been filled. Thus, the normal pattern in simple sentences (3a-d) appears also in (4a), but contrasts with (4b),

(4a) Dog who chases cat sees girl.

(4b) Dog who cat chases sees girl.

On the other hand, sentence (4c), which seems to conform to the pattern established in (3) and (4a), is ungrammatical.

(4c) *Dog who cat chases dog sees girl.

Similar complications arise for the agreements facts. In simple declarative sentences agreement involves *N1 - V1*. In complex sentences, such as (5a), that regularity is violated, and any straightforward attempt to generalize it to sentences with multiple clauses would lead to the ungrammatical (5b).

(5a) Dog who boys feed sees girl.

(5b) *Dog who boys feeds see girl.

(d) *Recursion*

The grammar permits recursion through the presence of relative clause (which expand to noun phrases which may introduce yet other relative clauses, etc.). This leads to sentences such as (6) in which the grammatical phenomena noted in (*a-c*) may be extended over a considerable distance.

(6) Boys who girls who dogs chase see hear.

(e) *Viable sentences*

One of the literals inserted by the grammar is ".", which occurs at the end of sentences. This end-of-sentence marker can potentially occur anywhere in a string where a grammatical sentence might be terminated. Thus in sentence (7), the carets indicate positions where a "." might legally occur.

(7) Boys see ^ dogs ^ who see ^ girls ^ who hear ^ .

* * *

The data in (4-7) are examples of the sorts of phenomena which linguists argue cannot be accounted for without abstract representations. More precisely, it has been claimed that such abstract representations offer a more perspicacious account of grammtical phenomena than one which, for example, simply lists the surface strings (Chomsky, 1957).

The training data were generated from the grammar summarized in Table 1. At any given point during training, the training set consisted of 10,000 sentences which were presented to the network 5 times. (As before, sentences were concatenated so that the input stream proceeded smoothly without breaks between sentences.) However, the composition of these sentences varied over time. The following training regimen was used in order to provide for incremental training. The network was trained on 5 passes through each of the following 4 corpora.

Phase 1: The first training set consisted exclusively of simple sentences. This was accomplished by eliminating all relative clauses. The result was a corpus of 34,605 words forming 10,000 sentences (each sentence includes the terminal ".").

Phase 2: The network was then exposed to a second corpus of 10,000 sentences which consisted of 25% complex sentences and 75% simple sentences (complex sentences were obtained by permitting relative clauses). Mean sentence length was 3.92 (minimum 3 words, maximum 13 words).

Phase 3: The third corpus increased the percentage of complex sentences to 50%, with mean sentence length of 4.38 (minimum: 3 words, maximum: 13 words).

Phase 4: The fourth consisted of 10,000 sentences, 75% complex, 25% simple. Mean sentence length was 6.02 (minimum: 3 words, maximum: 16 words).

This staged learning strategy was developed in response to results of earlier pilot work. In this work, it was found that the network was unable to learn the task when given the full range of complex data from the beginning of training. However, when the network was permitted to focus on the simpler data first, it was able to learn the task quickly and then move on successfully to more complex patterns. The important aspect to this was that the earlier training constrained later learning in a useful way; the early training forced the network to focus on canonical versions of the problems which apparently created a good basis for then solving the more difficult forms of the same problems.

Results

At the conclusion of the fourth phase of training, the weights were frozen at their final values and network performance was tested on a novel set of data, generated in the same way as the last training corpus. Because the task is non-deterministic, the network will (unless it memorizes the sequence) always produce errors. The optimal strategy in this case will be to activate the output units (i.e., predict potential next words) to some extent proportional to their statistical likelihood of occurrence. Therefore, rather than assessing the network's global performance by looking at root mean squared error, we should ask how closely the network approximated these probabilities. The technique described in Elman (in press) was used to accomplish this. Context-dependent likelihood vectors were generated for each word in every sentences; these vectors represented the empirically derived probabilities of occurrence for all possible predictions, given the sentence context up to that point. The network's actual outputs were then compared against these likelihood vectors, and this error was used to measure performance. The error was quite low: 0.177 (initial error: 12.45; minimal error through equal activation of all units would be 1.92). This error can also be normalized by computing the mean cosine of the angle between the vectors, which is 0.852 (sd: 0.259). Both measures indicate that the network achieved a high level of performance in prediction.

These gross measures of performance, however, do not tell us how well the network has done in each of the specific problem areas posed by the task. Let us look at each area in turn.

(a) *Agreement in simple sentences*

Agreement in simple sentences is shown in Figures 2a and 2b.

— Insert FIgure 2 about here —

The network's predictions following the word **boy** are that either a singular verb will follow (words in all three singular verb categories are activated, since it has no basis for predicting the type of verb), or else that the next word may be the relative pronoun **who**. Conversely, when the input is the word **boys**, the expectation is that a verb in the plural will follow, or else the relative pronoun. (Similar expectations hold for the other nouns in the lexicon. In this and the results that follow, the performance of the sentences which are shown is representative over other sentences with similar structure.)

(b) Verb argument structure in simple sentences

Figure 3 shows network predictions following an initial noun and then a verb from each of the three different verb types.

— Insert Figure 3 about here —

When the verb is **lives**, the network's expectation is that the following item will be "**.**" (which is in fact the only successor permitted by the grammar in this context). The verb **sees,** on the other hand, may either be followed by a "**.**", or optionally by a direct object (which may be a singular or plural noun, or proper noun). Finally, the verb **chases** requires a direct object, and the network learns to expect a noun following this and other verbs in the same class.

(c) Interactions with relative clauses

The examples so far have all involved simple sentences. The agreement and verb argument facts are more complicated in complex sentences. Figure 4 shows the network predictions for each word in the sentence **boys who mary chases feed cats.** If the network were generalizing the pattern for agreement found in the simple sentences, we might expect the network to predict a singular verb following **...mary chases...** (insofar as it predicts a verb in this position at all; conversely, it might be confused by the pattern *N1 N2 V1*). But in fact, the prediction (4d) is correctly that the next verb should be in the singular in order to agree with the first noun. In so doing, it has found some mechanism for representing the long-distance dependency between the main clause noun and main clause verb, despite the presence of an intervening noun and verb (with their own agreement relations) in the relative clause.

— Insert Figure 4 about here —

Note that this sentence also illustrates the sensitivity to an interaction between verb argument structure and relative clause structure. The verb **chases** takes an obligatory direct object. In simple sentences the direct object follows the verb immediately; this is also true in many complex sentences (e.g., **boys who chase mary feed cats**). In the sentence displayed, however, the direct object (**boys**) is the head of the relative clause and appears before the verb. This requires that the network learn (a) that there are items which function as nouns, verbs, etc.; (b) which items fall into which classes; (c) that there are subclasses of verbs which have different cooccurrence relations with nouns, corresponding to verb-direct object restrictions; (d) which verbs fall into which classes; and (e) when to expect that the direct object will follow the verb, and when to know that it has already appeared. The network appears to have learned this, because in panel (d) we see that it expects that **chases** will be followed by a verb (the main clause verb, in this case) rather than a noun.

An even subtler point is demonstrated in (4c). The appearance of **boys** followed by a relative clause containing a different subject (**who Mary**...) primes the network to expect that the verb which follows must be of the class that requires a direct object, precisely because a direct object filler has already appeared. In other words, the network correctly responds to the presence of a filler (**boys**) not only by knowing where to expect a gap (following **chases**); it also learns that when this filler corresponds to the object position in the relative clause, a verb is required which has the appropriate argument structure.

Network analysis

The natural question to ask at this point is how the network has learned to accomplish the task. Success on this task seems to constitute *prima facie* evidence for the existence of internal representations which possessed abstract structure. That is, it seemed reasonable to believe that in order to handle agreement and argument structure facts in the presence of relative clauses, the network would be required to develop representations which reflected constituent structure, argument structure, grammatical category, grammatical relations, and number. (At the very least, this is the same sort of inference which is made in the case of human language users, based on behavioral data.)

One advantage of working with an artificial system is that we can take the additional step of directly inspecting the internal mechanism which generates the behavior. Of course, the mechanism we find is is not necessarily that which is used by human listeners; but we may nonetheless be surprised to find solutions to the problems which we might not have guessed on our own.

Hierarchical clustering has been a useful analytic tool for helping to understand how the internatl representations which are learned by a network contribute to solving a problem. Clustering diagrams of hidden unit activation patterns is very good for representing the similarity structure of the representational space. However, it has certain limitations. One weakness is that it provides only an indirect picture of the representational space. Another shortcoming is that it tends to deemphasize the dynamics involved in processing. Some states may have significance not simply in terms of their similarity to other states, but with regard to the ways in which they constrain movement into subsequent state space (recall the examples in (1)). An important part of what the network has learned lies in the dynamics involved in processing word sequences. Indeed, one might think of the network dynamics as encoding grammatical knowledge; certain sequences of words move the network through well-defined and permissible internal states. Other sequences move the network through other permissible states. Some sequences are not permitted; these are ungrammatical.

What we might therefore wish to be able to do is directly inspect the internal states (represented by the hidden unit activation vectors) the network is in as it processes words in sequence, in order to see how the states and the trajectories encode the network's grammatical knowledge.

Unfortunately, the high dimensionality of the hidden unit activation vectors (in the simulation here, 70 dimensions) makes it impractical to view the state space directly. Furthermore, there is no guarantee that the dimensions which will be of interest to us—in the sense that they pick out regions of importance in network's solution to the task—will be correlated with any of the dimensions coded by the hidden units. Indeed, this is what it means for the representations to be distributed: the dimensions of variation cut across, to some degree, the dimensions picked out by the hidden units.

However, it is reasonable to assume that such dimensions of variation do exist, we can try to identify them using principal component analysis (PCA). PCA allows us to find another set of dimensions (a rotation of the axes) along which maximum variation occurs[1]. (It may additionally reduce the number of variables by effectively removing the linearly dependent set of axes). These new axes permit us to visualize the state space in a way which hopefully allows us to see how the network solves the task. (A shortcoming of PCA is that it is linear; however, the combination of the PCA factors at the next level may be non-linear, and so this representation of information may give an incomplete picture of the actual computation.) Each dimension (eigenvector) has an associated eigenvalue, the magnitude of which indicates the amount of variance accounted for by that dimension. This allows one to focus on dimensions which may be of particular significance; it also allows a *post hoc* estimate of the number of hidden units which might actually be required for the task. Figure 5 shows a graph of the eigenvalues of the 70 eigenvectors which were extracted.

— Insert Figure 5 about here —

<u>Agreement</u>

The sentences in (8) were presented to the network, and the hidden unit patterns captured after each word was processed in sequence.

(8a) boys hear boys .
(8b) boy hears boys .
(8c) boy who boys chase chases boy .
(8d) boys who boys chase chase boy .

(These sentences were chosen to minimize differences due to lexical content and to make it possible to focus on differences to grammatical structure. (8a) and (8b) were contained in the training data; (8c) and (8d) were novel and had never been presented to the network during learning.)

By examining the trajectories through state space along various dimensions, it was apparent that the second principal component played an important role in marking number of the main clause subject. Figure 6 shows the trajectories for (8a) and (8b); the trajectories are overlaid so that the differences are more readily seen. The paths are similar

---

[1]In practical terms, this anaylsis involves passing the training set through the trained network (with weights frozen) and saving the hidden unit patterns that are produced in response to each input. The covariance matrix of the resulting set of hidden unit vectors is calculated, and then the eigenvectors of the covariance matrix are found. The eigenvectors are ordered by the magnitude of their eigenvalues, and are used as the basis for describing the original hidden unit vectors. This new set of dimensions has the effect of giving a somewhat more localized description to the hidden unit patterns, because the new dimensions now correspond to the location of meaningful activity (defined in terms of variance) in the hyperspace. Since the dimensions are ordered in terms of variance accounted for, we may wish to look at selected dimensions, starting with those with largest eigenvalues. See Flury (1988) for a detailed explanation of PCA; or Gonzalez & Wintz (1977) for a detailed description of the algorithm.

and diverge only during the first word, indicating the difference in the number of the initial noun. The difference is slight and is eliminated after the main (i.e., second **chase**) verb has been input. This is apparently because, for these two sentences (and for the grammar), number information does not have any relevance for this task once the main verb has been received.

— Insert Figure 6 about here —

It is not difficult to imagine sentences in which number information may have to be retained over an intervening constituent; sentences (8c) and (8d) are such examples. In both these sentences there is an identical relative clause which follows the initial noun (which differs with regard to number in the two sentences). This material, **who boys chase**, is irrelevant as far as the agreement requirements for the main clause verb. The trajectories through state space for these two sentences have been overlaid and are shown in Figure 7; as can be seen, the differences in the two trajectories are maintained until the main clause verb is reached, at which point the states converge.

— Insert Figure 7 about here —

Verb argument structure

The representation of verb argument structure was examined by probing with sentences containing instances of the three different classes of verbs. Sample sentences are shown in (9).

(9a) boy walks .
(9b) boy sees boy .
(9c) boy chases boy .

The first of these contains a verb which may not take a direct object; the second takes an option direct object; and the third requires a direct object. The movement through state space as these three sentences are processed are shown in Figure 8.

— Insert Figure 8 about here —

This figure illustrates how the network encodes several aspects of grammatical structure. Nouns are distinguished by role; subject nouns for all three sentences appear in the upper right portion of the space, and object nouns appear below them. (Principal component 4, not shown here, encodes the distinction between verbs and nouns, collapsing across case.) Verbs are differentiated with regard to their argument structure. **Chases** requires a direct object, **sees** takes an optional direct object, and **walks** precludes an object. The difference is reflected in a systematic displacement in the plane of principal components 1 and 3.

Relative clauses

The presence of relative clauses introduces a complication into the grammar, in that the representations of number and verb argument structure must be clause-specific.

It would be useful for the network to have some way to represent the constituent structure of sentences.

The trained network was given the following sentences.

(10a) boy chases boy .
(10b) boy chases boy who chases boy .
(10c) boy who chases boy chases boy .
(10d) boy chases boy who chases boy who chases boy .

The first sentence is simple; the other three are instances of embedded sentences. Sentence10a was contained in the training data; sentences 10c, 10d, and 10e were novel and had not been presented to the network during the learning phase.

The trajectories through state space for these four sentences (principal components 1 and 11) are shown in Figure 9. Panel (9a) shows the basic pattern associated with what is in fact the matrix sentences for all four sentences. Comparison of this figure with panels (9b) and (9c) shows that the trajectory for the matrix sentence appears to follow the same for; the matrix subject noun is in the lower left region of state space, the matrix verb appears above it and to the left, and the matrix object noun is near the upper middle region. (Recall that we are looking at only 2 of the 70 dimensions; along other dimensions the noun/verb distinction is preserved categorically.) The relative clause appears involve a replication of this basic pattern, but displaced toward the left and moved slightly downward, relative to the matrix constituents. Moreover, the exact position of the relative clause elements indicates which of the matrix nouns are modified Thus, the relative clause modifying the subject noun is closer to it, and the relative clause modifying the object noun are closer to it. This trajectory pattern was found for all sentences with the same grammatical form; the pattern is thus systematic.

— Insert Figure 9 about here —

Figure (9d) shows what happens when there are multiple levels of embedding. Successive embeddings are represented in a manner which is similar to the way that the first embedded clause is distinguished from the main clause; the basic patter for the clause is replicated in region of state space which is displaced from the matrix material. This displacement provides a systematic way for the network to encode the depth of embedding in the current state. However, the reliability of the encoding is limited by the precision with which states are represented, which in turn depends on factors such as the number of hidden units and the precision of the numerical values. In the current simulation, the representation degraded after about three levels of embedding. The consequences of this degradation on performance (in the prediction task) are different for different types of sentences. Sentences involving center embedding (e.g., 9c and 9d), in which the level of embedding is crucial for maintaining correct agreement, are more adversely affected than sentences involving so-called tail-recursion (e.g., 10d). In these latter sentences the syntactic structures in principle involve recursion, but in practice the level of embedding is not relevant for the task (i.e., does not affect agreement or verb argument structure in any way).

Figure 9d is interesting in another respect. Given the nature of the prediction task, it is actually not necessary for the network to carry forward any information from prior clauses. It would be sufficient for the network to represent each successive relative clause as an iteration of the previous pattern. Yet the two relative clauses are differentiated. Similarly, Servan-Schreiber, Cleeremans, & McClelland (in press) found that when a simple recurrent network was taught to predict inputs that had been generated by a finite state automaton, the network developed internal representations which corresponded to the FSA states; however, it also redundantly made finer-grained distinctions which encoded the path by which the state had been achieved, even though this information was not used for the task. It thus seems to be a property of these networks that while they are able to encode state in a way which minimizes context as far as behavior is concerned, their nonlinear nature allows them to remain sensitive to context at the level of internal representation.

Discussion

The basic question addressed in this paper is whether or not connectionist models are capable of complex representations which possess internal structure and which are productively estensible. This question is particularly of interest with regards to a more general issue: How useful is the connectionist paradigm as a framework for cognitive models? In this context, the nature of representations interacts with a number of other closely related issues. So in order to understand the significance of the present results, it may be useful first to consider briefly two of these other issues. The first is the status of *rules* (whether they exist, whether they are explicit or implicit); the second is the notion of *computational power* (whether it is sufficient, whether it is appropriate).

It is sometimes suggested that connectionist models differ from Classical models in that the latter rely on rules whereas connectionist models are typically not rule systems. Although at first glance this appears to be a reasonable distinction, it is not actually clear that the distinction gets us very far.

The basic problem is that it is not obvious what is meant by a rule. In the most general sense, a rule is a mapping which takes an input and yields an output. Clearly, since many (although not all) neural networks function as input/output systems in which the bulk of the machinery implements some transformation, it is difficult to see how they could not be thought of as rule-systems.

But perhaps what is meant is that the *form* of the rules differs in Classical models and connectionist networks? One suggestion has been that rules are stated *explicitly* in the former, whereas they are only *implicit* in networks. This is a slippery issue, and there is an unfortunate ambiguity in what is meant by implicit or explicit.

One sense of explicit is that a rule is physically present in the system *in its form as a rule*; and furthermore, that that physical presence is important to the correct functioning of the system. However, Kirsh (1989) points out that our intuitions as to what counts as physical presence are highly unreliable and sometimes contradictory. What seems to really be at stake is the speed with which information can be made available. If this is true, and Kirsh argues the point persuasively, then the quality of explicitness does not belong to data structures alone. One must also take into account the nature of the processing system involved, since information in the same form may be easily accessible in one processing system and inaccessible in another.

Unfortunately, our understanding of the information processing capacity of neural networks is quite preliminary. There is a strong tendency in analyzing such networks to view them through traditional lenses. We suppose that if information is not contained in the same form as more familiar computational systems, that information is somehow buried, inaccessible, and implicit. Consider, for instance, a network which successfully learns some complicated mapping — say, from text to pronunciation (Sejnowski & Rosenberg, 1987. On inspecting the resulting network, it is not immediately obvious how to explain how the mapping works or even to characterize what the mapping is in any precise way. In such cases, it is tempting to say that the network has learned an implicit set of rules. But what we really mean is just that the mapping is "complicated", or "difficult to formulate", or even "unknown". This is rather a description of our own failure to understand the mechanism rather than a description of the mechanism itself. What is needed are new techniques for network analysis, such as the principal component analysis used in the present work, contribution analysis (Sanger, 1989), weight matrix decomposition (McMillan & Smolensky, 1988), or skeletonization (Mozer & Smolensky, 1989).

If successful, these analyses of connectionist networks may provide us with a new vocabulary for understanding information processing. We may learn new ways in which information can be explicit or implicit, and we may learn new notations for expressing the rules that underlie cognition. The notation of these new connectionist rules may look very different than that used in, for example, production rules. And we may expect that the notation will not lend itself to describing all types of regularity with equal facility.

Thus, the potential important difference between connectionist models and Classical models will not be in whether one or the other systems contains rules, or whether one system encodes information explicitly and the other encodes it implicitly; the difference will lie in the nature of the rules, and in what kinds of information count as explicitly present.

This potential difference brings us to the second issue: computational power. The issue divides into two considerations. Do connectionist models provide *sufficient* computational power (to account for cognitive phenomena); and do they provide the *appropriate* sort of computational power?

The first question can be answered affirmatively with an important qualification. It can be shown that multilayer feedforward networks with as few as one hidden layer, with no squashing at the output and an arbitrary nonlinear activation function at the hidden layer, are capable of arbitrarily accurate approximation of arbitrary mappings. They thus belong to a class of universal approximators (Hornik, Stinchcombe, & White, in press; Stinchcombe & White, 1989). Pollack (1988) has also proven the Turing equivalence of neural networks. In principle, then, such networks are capable of implementing any function that the Classical system can implement.

The important qualification to the above results is that sufficiently many hidden units be provided (or in the case of Pollack's proof, that weights be infinite precision). What is not currently known is effect of limited resources on computational power. Since human cognition is carried out in a system with relatively fixed and limited resources, this question is of paramount interest. These limitations provide critical constraints on the nature of the functions which can be mapped; it is an important empirical question whether these constraints explain the specific form of human cognition.

It is in this context that the question of the appropriateness of the computational power becomes interesting. Given limited resources, it is relevant to ask whether the kinds of operations and representations which are naturally made available are those which are likely to figure in human cognition. If one has a theory of cognition which requires sorting of randomly ordered information, e.g., word frequency lists in Forster's (1979) model of lexical access, then it becomes extremely important that the computational framework provide efficient support for the sort operation. On the other hand, if one believes that information is stored associatively, then the ability of the system to do a fast sort is irrelevant. Instead, it is important that the model provide for associative storage and retrieval[1]. Of course, things work in both directions. The availability of certain types of operations may encourage one to build models of a type which are impractical in other frameworks. And the need to work with an inappropriate computational mechanism may blind us from seeing things as they really are.

* * * *

Let us return now to the current work. I would like to discuss first some of the ways in which the work is preliminary and limited. Then I will discuss what I see as the positive contributions of the work. Finally, I would like to relate this work to other connectionist research and to the general question raised at the outset of this discussion: How viable are connectionist models for understanding cognition?

The results are preliminary in a number of ways. First, one can imagine a number

[1] This example was suggested to me by Don Norman.

of additional tests that could be performed to test the representational capacity of the simple recurrent network. The memory capacity remains unprobed (but see Servan-Schreiber, Cleeremans, & McClelland, in press). Generalization has been tested in a limited way (many of the tests involved novels sentences), but one would like to know whether the network can inferentially extend what it knows about the types of noun phrases encountered in the second simulation (simple nouns and relative clauses) to noun phrases with different structures.

Second, while it is true that the agreement and verb argument structure facts contained in the present grammar are important and challenging, we have barely scratched the surface in terms of the richness of linguistic phenomena which characterize natural languages.

Third, natural languages not only contain far more complexity with regard to their syntactic structure, they also have a semantic aspect. Indeed, Langacker (1987) and others have argued persuasively that it is not fruitful to consider syntax and semantics as autonomous aspects of language. Rather, the form and meaning of language are closely entwined. Although there may be things which can be learned by studying artificial languages such as the present one which are purely syntactic, *natural* language processing is crucially an attempt to retrieve meaning from linguistic form. The present work does not address this issue at all, but there are other PDP models which have made progress on this problem (e.g., St. John & McClelland, in press).

What the current work does contribute is some notion of the representational capacity of connectionist models. Various writers (e.g., Fodor & Pylyshyn, 1988) have expressed concern regarding the ability of connectionist representations to encode compositional structure and to provide for open-ended generative capacity. The networks used in the simulations reported here have two important properties which are relevant to these concerns.

First, the networks make possible the development of internal representations that are *distributed* (Hinton, 1988; Hinton, McClelland, Rumelhart, 1986). While not unbounded, distributed representations are less rigidly coupled with resources than localist representations, in which there is a strict mapping between concept and individual nodes.. There is also greater flexibility in determining the dimensions of importance for the model.

Second, the networks studied here build in a sensitivity to context. The important result of the current work is to suggest that the sensitivity to context which is characteristic of many connectionist models, and which is built-in to the architecture of the networks used here, does not preclude the ability to capture generalizations which are at a high level of abstraction. Nor is this a paradox. Sensitivity to context is precisely the mechanism which underlies the ability to abstract and generalize. The fact that the networks here exhibited behavior which was highly regular was not because they learned to be context-insensitive. Rather, they learned to respond to contexts which are more abstractly defined. Recall that even when these networks' behavior seems to ignore context (e.g., Figure 9d; and Servan-Schreiber, Cleeremans, & McClelland, in press), the internal representations reveal that contextual information is still retained.

This behavior is in striking contrast to that of traditional symbolic models. Representations in these systems are naturally context-*insensitive*. This insensitivity makes it possible to express generalizations which are fully regular at the highest possible level of representation (e.g., purely syntactic), but they require additional apparatus to account for regularities which reflect the interaction of meaning with form and which are more contextually defined. Connectionist models on the other hand begin the task of abstraction at the other end of the continuum. They emphasize the importance of context and the interaction of form with meaning. As the current work demonstrates, these characteristics lead quite naturally to generalizations at high level of abstraction where appropriate, but the behavior remains ever-rooted in representations which are contextually grounded. The simulations reported here do not capitalize on subtle distinctions in context, but there are ample demonstrations of models which do (e.g., Kawamoto, 1988; McClelland & Kawamoto, 1986; Miikkulainen & Dyer, 1989; St. John & McClelland, in press).

Finally, I wish to point out that the current approach suggests a novel way of thinking about how mental representations are constructed from language input.

Conventional wisdom holds that as words are heard, listeners retrieve lexical representations. Although these representations may indicate the contexts in which the words acceptably occur, the representations are themselves context-free. They exist in some canonical form which is constant across all occurrences. These lexical forms are then used to assist in constructing a complex representation into which the forms are inserted. One can imagine that when complete, the result is an elaborate structure in which not only are the words visible, but which also depicts the abstract grammatical structure which binds those words.

In this account, the process of building mental structures is not unlike the process of building any other physical structure, such as bridges or houses. Words (and whatever other representational elements are involved) play the role of building blocks. As is true of bridges and houses, the building blocks are themselves unaffected by the process of construction.

A different image is suggested in the approach taken here. As words are processed there is no separate stage of lexical retrieval. There are no representations of words in isolation. The representations of words (the internal states following input of a word) always reflect the input taken together with the prior state. In this scenario, words are not building blocks as much as they are cues which guide the network through different grammatical states. Words are distinct from each other by virtue of having different causal properties.

A metaphor which captures some of the characteristics of this approach is the combination lock. In this metaphor, the role of words is analogous to the role played by the numbers in the combination. The numbers have causal properties; they advance the lock into different states. The effect of a number is dependent on its context. Entered in the correct sequence, the numbers move the lock into an open state. The open state may be said to be *functionally compositional* (van Gelder, in press) in the sense that it reflects a particular sequence of events. The numbers are "present" insofar as they are responsible for the final state, but not because they are still physically present.

The limitation of the combination lock is of course that there is only one correct combination. The networks studied here are more complex. The causal properties of the words are highly structure-dependent and the networks allow many "open" (i.e., grammatical) states.

This view of language comprehension emphasizes the functional importance of representations and is similar in spirit to the approach described in Bates & MacWhinney, 1982; McClelland, St. John, & Taraban, 1989; and many others who have stressed the functional nature of language. Representations of language are constructed in order to accomplish some behavior (where, obviously, that behavior may range from day-dreaming to verbal duels, and from to asking directions to composing poetry). The representations are not propositional, and their information content changes constantly over time in accord with the demands of the current task. Words serve as guideposts which help establish mental states that support this behavior; representations are snapshots of those mental states.

ACKNOWLEDGMENTS

I am grateful for many useful discussions on this topic with Jay McClelland, Dave Rumelhart, Elizabeth Bates, Steve Stich, and members of the UCSD PDP/NLP Research Group. I thank McClelland, Mike Jordan, Mary Hare, Ken Baldwin, and two anonymous reviewers for critical comments on earlier versions of this paper. This research was supported by contracts N00014-85-K-0076 from the Office of Naval Research and contract DAAB-07-87-C-H027 from Army Avionics, Ft. Monmouth. Requests for reprints should be sent to the Center for Research in Language, 0126; University of California, San Diego; La Jolla, CA 92093-0126. The author can be reached via electronic mail as elman@crl.ucsd.edu.

REFERENCES

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

Bates, E., & MacWhinney, B. (1982). Functionalist approaches to grammar. In E. Wanner, & L. Gleitman (Eds.), *Language acquisition: The state of the art.* New York: Cambridge University Press.

Chafe, W. (1970). *Meaning and the Structure of Language.* Chicago: University of Chicago Press.

Chalmers, D.J. (1990). Syntactic transformations on distributed representations. Center for Research on Concepts and Cognition, Indiana University.

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

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

Dolan, C., & Dyer, M.G. (1987). Symbolic schemata in connectionist memories: Role binding and the evolution of structure. Technical Report UCLA-AI-87-11. Artificial Intelligence Laboratory, University of California, Los Angeles.

Dolan, C.P., & Smolensky, P. (1988). Implementing a connectionist production system using tensor products. Technical Report UCLA-AI-88-15, Artificial Intelligence Laboratory, University of California, Los Angeles.

Elman, J.L. (1989). Representation and structure in connectionist models. Technical Report CRL-8903. Center for Research in Language, University of California, San Diego.

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

Fauconnier, G. (1985). *Mental Spaces.* Cambridge, MA: MIT Press.

Feldman, J. A. & Ballard, D. H., 1982. Connectionist models and their properties. *Cognitive Science, 6,* 205-254.

Fillmore, C.J. (1982). Frame semantics. In *Linguistics in the Morning Calm.* Seoul: Hansin.

Flury, B. (1988). *Common principal components and related multivariate models.* New York: Wiley.

Fodor, J. (1976). *The language of thought.* Harvester Press, Sussex.

Fodor, J., & Pylyshyn, Z. (1988). Connectionism and cognitive architecture: A critical analysis. In S. Pinker & J. Mehler (Eds.), *Connections and Symbols* . Cambridge, MA: MIT Press.

Forster, K.I. (1979). Levels of processing and the structure of the language processor. In W.E. Cooper & E. Walker (Eds.) *Sentence processing: Psycholinguistic studies presented to Merrill Garrett.* Hillsdale, N.J.: Lawrence Erlbaum Associates.

Gasser, M., & Lee, C-D. (1990). Networks that learn phonology. Computer Science Department, Indiana University.

Givon, T. (1984). *Syntax: A Functional-Typological Introduction. Volume 1.* Amsterdam: John Benjamins.

Gold, E.M. (1967). Language identification in the limit. *Information and Control, 16,* 447-474.

Gonzalez, R.C., & Wintz., P. (1977). *Digital Image Processing.* Reading, MA: Addison-Wesley.

Grosjean, F. (1980). Spoken word recognition processes and the gating paradigm. *Perception & Psychophysics, 28,* 267-283.

Hanson, S.J., & Burr, D.J. (1987). Knowledge representation in connectionist networks. Bell Communications Research, Morristown, New Jersey.

Hare, M. (1990). The role of similarity in Hungarian vowel harmony: A connectionist account. CRL Technical Report 9004. Center for Research in Language, University of California, San Diego.

Hare, M., Corina, D., & Cottrell, G. (1988) Connectionist perspective on prosodic structure. CRL Newsletter, Vol. 3, No. 2. Center for Research in Language, University of California, San Diego.

Hinton, G.E. (1988). Representing part-whole hierarchies in connectionist networks. Technical Report CRG-TR-88-2, Connectionist Research Group, University of Toronto.

Hinton, G.E., McClelland, J.L., & Rumelhart, D.E. (1986). Distributed representations. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition (Vol. 1)* Cambridge, MA: MIT Press.

Hopper, P.J., & Thompson, S.A. (1980). Transitivity in grammar and discourse. *Language, 56,* 251-299.

Hornik, K., Stinchcombe, M., & White, H. (in press). Multi-layer feedforward networks are universal approximators. *Neural Networks.*

Jordan, M. I. (1986). Serial order: A parallel distributed processing approach. Institute for Cognitive Science Report 8604. University of California, San Diego.

Kawamoto, A.H. (1988). Distributed representations of ambiguous words and their resolution in a connectionist network. In S.L. Small, G.W. Cottrell, & M.K. Tanenhaus (Eds.), *Lexical ambiguity resolution: Perspectives from psycholinguistics, neuropsychology, and artificial intelligence.* San Mateo, CA: Morgan Kaufmann Publishers.

Kirsh, D. (in press). When is information represented explicitly? In J. Hanson (Ed.), *Information, Thought, and Content.* Vancouver: University of British Columbia.

Kuno, S. (1987). *Functional syntax: Anaphora, discourse and empathy.* Chicago: The University of Chicago Press.

Kutas, M. (1988). Event-related brain potentials (ERPs) elicited during rapid serial presentation of congruous and incongruous sentences. in R. Rohrbaugh, J. Rohrbaugh, & P. Parasuramen (Eds.), *Current Trends in Brain Potential Research (EEG Supplement 40).* Amsterdam: Elsevier.

Kutas, M., & Hillyard, S.A. (1980). Reading senseless sentences: Brain potentials reflect semantic incongruity. *Science,* 207, 203-205.

Lakoff, G. (1987). *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind.* Chicago: University of Chicago Press.

Langacker, R.W. (1987). *Foundations of Cognitive Grammar: Theoretical Perspectives. Volume 1.* Stanford: Stanford University Press.

Langacker, R.W. (1988). A usage-based model. *Current Issues in Linguistic Theory, 50,* 127-161.

MacWhinney, B., Leinbach, J., Taraban, R., & McDonald, J. (1989). Language learning: Cues or rules? *Journal of Memory and Language, 28,* 255-277.

Marslen-Wilson, W., & Tyler, L.K. (1980). The temporal structure of spoken language understanding. *Cognition, 8,* 1-71

McClelland, J.L. (1987). The case for interactionism in language processing. In M. Coltheart (Ed.), *Attention and performance XII: The psychology of reading.* London: Erlbaum.

McClelland, J.L., St. John, M., & Taraban, R. (1989). Sentence comprehension: A parallel distributed processing approach. Manuscript. Department of Psychology, Carnegie Mellon University..

McMillan, C., & Smolensky, P. (1988). Analyzing a connectionist model as a system of soft rules. Technical Report CU-CS-303-88, Department of Computer Science, University of Colorado, Boulder.

Miikkulainen, R., & Dyer, M. (1989a). Encoding input/output representations in connectionist cognitive systems. In D.S. Touretzky, G.E. Hinton, & T.J. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School.* Los Altos, CA: Morgan Kaufmann Publishers.

Miikkulainen, R., & Dyer, M. (1989b). A modular neural network architecture for sequential paraphrasing of script-based stories. In *Proceedings of the International Joint Conference on Neural Networks,* IEEE.

Mozer, M. (1988). A focused back-propagation algorithm for temporal pattern recognition. Technical Report CRG-TR-88-3, Departments of Psychology and Computer Science, University of Toronto.

Mozer, M.C., & Smolensky, P. (1989). Skeletonization: A technique for trimming the fat from a network via relevance assessment. Technical Report CU-CS-421-89, Department of Computer Science, University of Colorado, Boulder.

Oden, G. (1978). Semantic constraints and judged preference for interpretations of ambiguous sentences. *Memory and Cognition, 6,* 26-37.

Pinker, S. (1989). *Learnability and Cognitiion: The Acquisition of Argument Structure.* Cambridge, MA: MIT Press.

Pollack, J.B. (1988). Recursive auto-associative memory: Decising compositional distributed representations. *Proceedings of the Tenth Annual Conference of the Cognitive Science Society.* Hillsdale, N.J.: Lawrence Erlbaum

Pollack, J.B. (in press). Recrusive distributed representations. *Artificial Intelligence.*

Ramsey, W. (1989). *The philosophical implications of connectionism.* Ph.D. thesis, University of California, San Diego.

Reich, P.A>, & Dell, G.S. (1977). Finiteness and embedding. In E.L. Blansitt, Jr., & P. Maher (Eds.), *The Third LACUS Forum*. Columbia, SC: Hornbeam Press.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning internal representations by error propagation. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition (Vol. 1)*. Cambridge, MA: MIT Press.

Rumelhart, D.E>, & McClelland, J.L. (1986a). PDP Models and general issues in cognitive science. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition (Vol. 1)*. Cambridge, MA: MIT Press.

Rumelhart, D.E>, & McClelland, J.L. (1986b). On learning the past tenses of English verbs. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition (Vol. 1)*. Cambridge, MA: MIT Press.

Salasoo, A., & Pisoni, D.B. (1985). Interaction of knowledge sources in spoken word identification. *Journal of Memory and Language, 24,* 210-231.

Sanger, D. (1989). Contribution analysis: A technique for assigning responsibilities to hidden units in connectionist networks. Technical Report CU-CS-435-89, Department of Computer Science, University of Colorado, Boulder.

Schlesinger, I.M. (1971). On linguistic competence. In Y. Bar-Hillel (Ed.), *Pragmatics of Natural Languages.* Dordrecht, Holland: Reidel.

Sejnowski, T.J., & Rosenberg, C.R. (1987). Parallel networks that learn to pronounce English text. *Complex Systems, 1,* 145-168.

Servan-Schreiber, D., Cleeremans, A., & McClelland, J.L. (in press). Graded state machines: The representation of temporal contingencies in simple recurrent networks. *Machine Learning: Special Issue on Neural Networks.*

Shastri, L., & Ajjanagadde, V. (1989). A connectionist system for rule based reasoning with multi-place predicates and variables. Technical Report MS-CIS-8905, Computer and Information Science Department, University of Pennsylvania.

Smolensky, P. (1987a). On variable binding and the representation of symbolic structures in connectionist systems. Technical Report CU-CS-355-87, Department of Computer Science, University of Colorado, Boulder.

Smolensky, P. (1987b). On the proper treatment of connectionism. Technical Report CU-CS-377-87, Department of Computer Science, University of Colorado, Boulder.

Smolensky, P. (1987c). Putting together connectionism - again. Technical Report CU-CS-378-87, Department of Computer Science, University of Colorado, Boulder.

Smolensky, P. (1988). On the proper treatment of connectionism. *The Behavioral and Brain Sciences, 11.*

Smolensky, P. (in press). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence.*

St. John, M., & McClelland, J.L. (in press). Learning and applying contextual constraints in sentence comprehension. Technical Report. Department of Psychology. Carnegie-Mellon University.

Stemberger, J.P. (1985). *The lexicon in a model of language production.* New York: Garland Publishing.

Stinchcombe, M., & White, H. (1989). Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions. *Proceedings of the International Joint Conference on Neural Networks,* Washington, D.C.

Stolz, W. (1967). A study of the ability to decode grammatically novel sentences. *Journal of Verbal Learning and Verbal Behavior,* 6, 867-873.

Tanenhaus, M.K., Garnseyh, S.M., & Boland, J. (in press). Combinatory lexical information and language comprehension. In G. Altmann (Ed.), *Cognitive Models of Speech Processing: Psycholinguistic and Computational Perspectives.* Cambridge, MA: MIT Press.

Touretzky, D.S. (1986). BoltzCONS: Reconciling connectionism with the recursive nature of stacks and trees. *Proceedings of the Eight Annual Conference of the Cognitive Science Society.* Hillsdale, N.J.: Lawrence Erlbaum.

Touretzky, D.S. (1989). Rules and maps in connectionist symbol processing. Technical Report CMU-CS-89-158, Department of Computer Science, Carnegie-Mellon University.

Touretzky, D.S. (1989). Towards a connectionist phonology: The “many maps” approach to sequence manipulation. *Proceedings of the 11th Annual Conference of the Cognitive Science Society,* 188-195.

Touretzky, D.S., & Hinton, G.E. (1985). Symbols among the neurons: Details of a connectionist inference architecture. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Los Angeles.*

Touretzky, D.S., & Wheeler, D.W. (1989). A connectionist implementation of cognitive phonology. Technical Report CMU-CS-89-144. Pittsburgh: Carnegie Mellon University, School of Computer Science.

Van Gelder, T.J. (in press). Compositionality: Variations on a classical theme. *Cognitive Science.*

FIGURE LEGENDS

1. Network architecture. Hidden unit activations are copied along fixed weights (of 1.0) into linear Context units on a one-to-one basis; on the next time step the Context units feed into Hidden units on a distributed basis. Additional hidden units between input and main hidden layer, and between main hidden layer and output, provide compress basis vectors into more compact form.

2. (a) Graph of network predictions following presentation of the word **boy**. Predictions are shown as activations for words grouped by category. **S** stands for end-of-sentence ("."); **W** stands for who; **N** and **V** represent nouns and verbs; **1** and **2** indicate singular or plural; and type of verb is indicated by **N**, **R**, **O** (direct object no possible, required, or optional). (b) Graph of network predictions following presentation of the word **boys**.

3. Graph of network predictions following the sequences **boy lives ...** ; **boy sees ...** ; and **boy chases ...** (the first precludes a direct object, the second optional permits a direct object, and the third requires a direct object).

4. Graph of network predictions after each word in the sentence **boys who mary chases feed dogs .** is input.

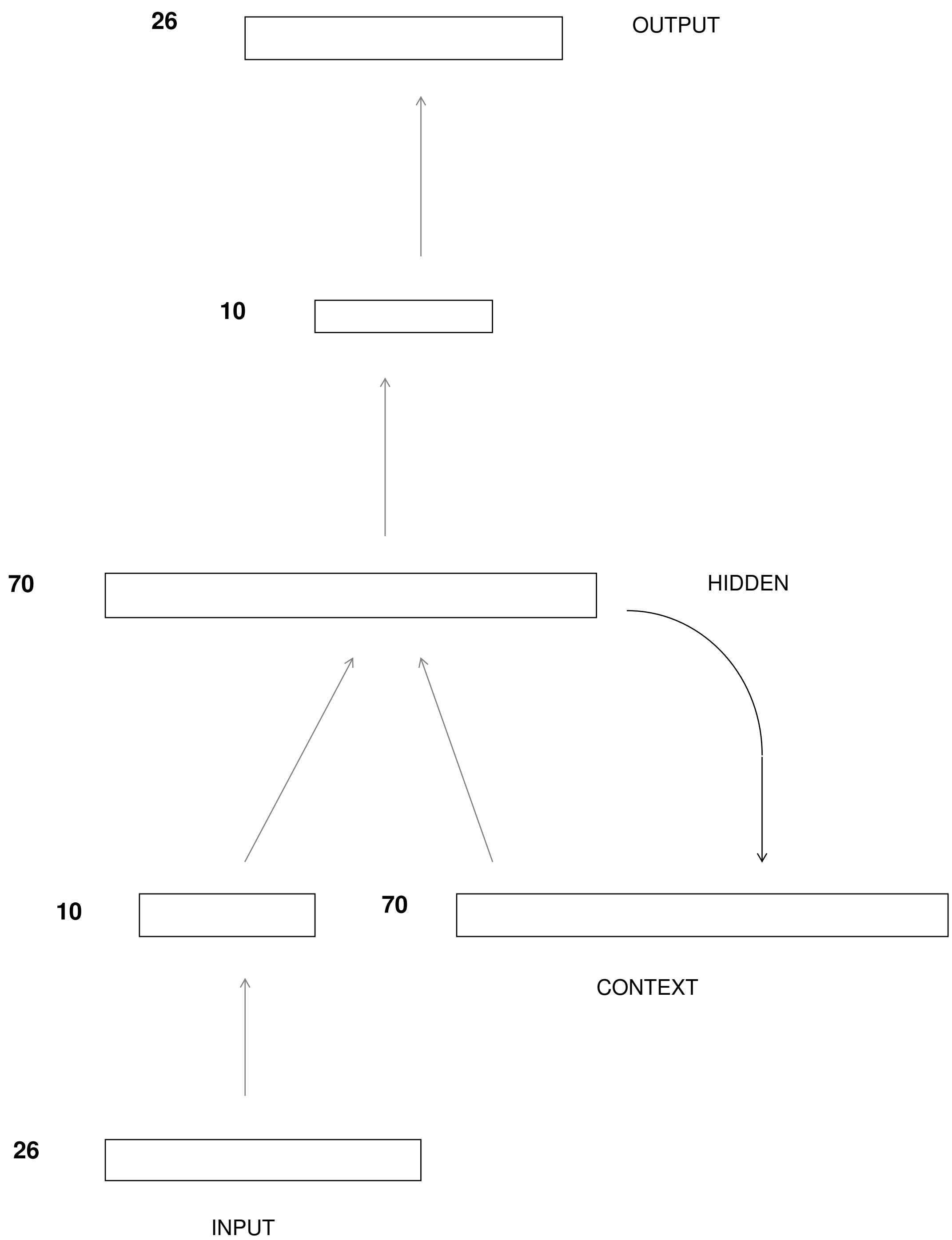
5. Graph of eigenvalues of the 70 ordered eigenvectors extracted in Simulation 2.

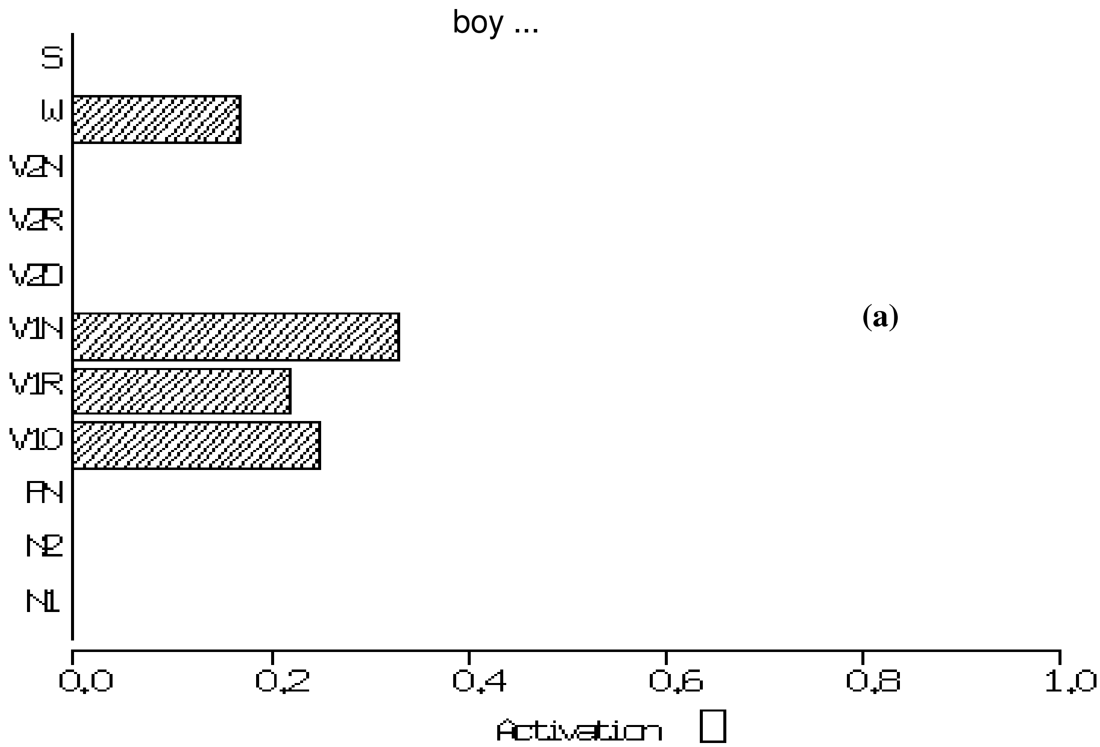
6. Trajectories through state space for sentences (8a) and (8b). Each point marks the position along the second principle component of hidden units space, after the indicated word has been input. Magnitude of the second principle component is measured along the ordinate; time (i.e., order of word in sentence) is measured along the abscissa. In this and subsequent graphs the sentence-final word is marked with a **]S** .

7. Trajectories through state space during processing of (8c) and (8d).

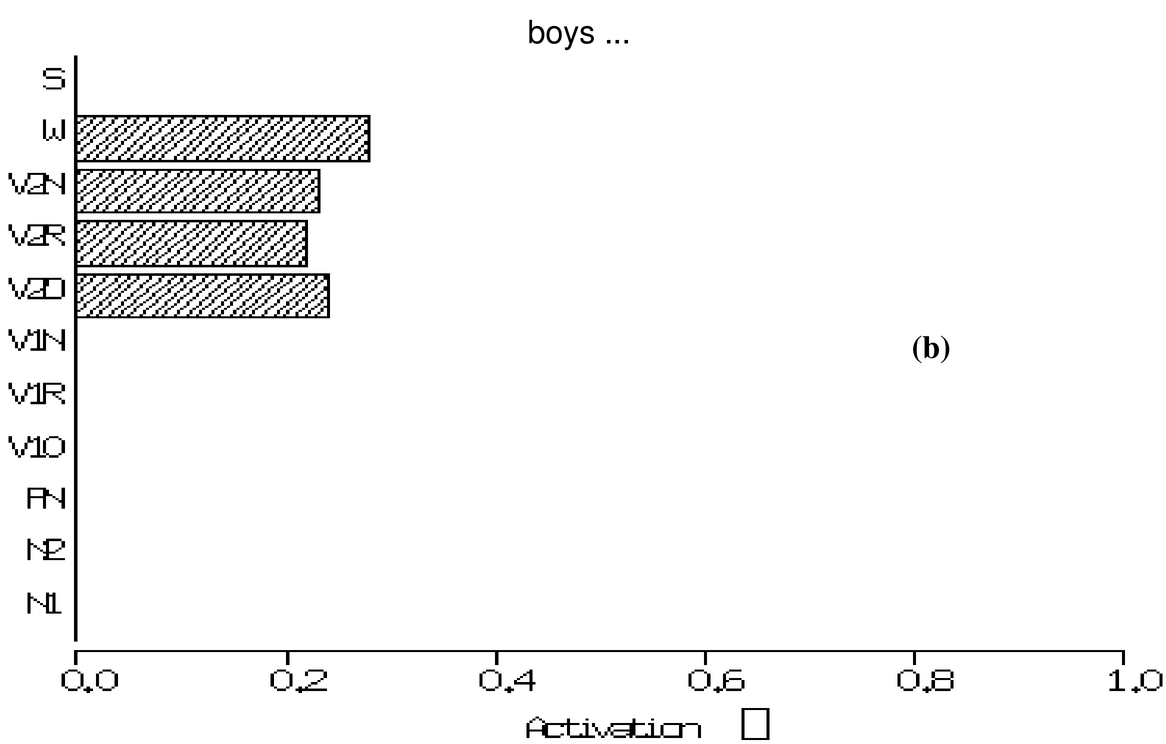
8. Trajectories through state space for sentences (9a), (9b), and (9c). Principle component 1 is plotted along the abscissa; principal component 3 is plotted along the ordinate.

9. Trajectories through state space for sentences (10a-d). Principle component 1 is displayed along the abscissa; principal component 11 is plotted along the ordinate.


26
OUTPUT
10
70
HIDDEN
10
70
CONTEXT
26
INPUT

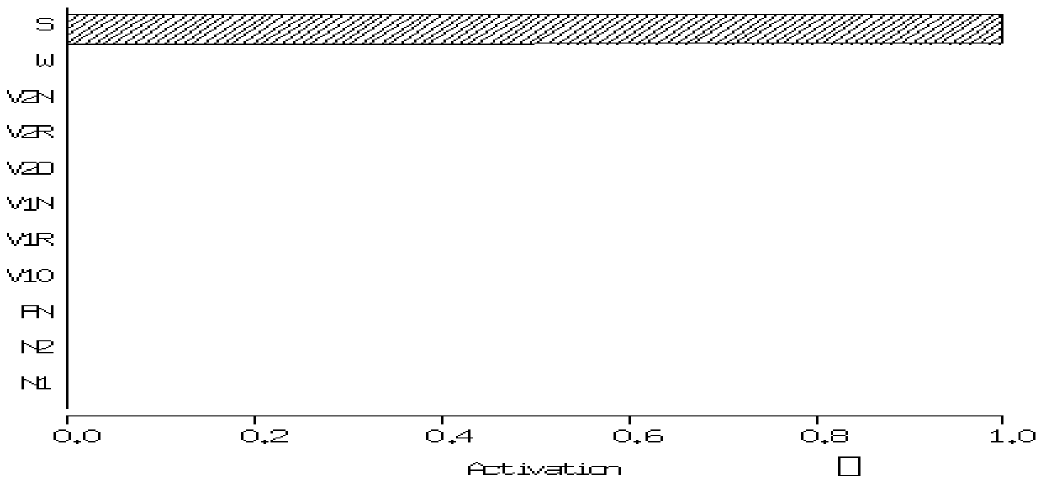

boy ...
S
W
V2N
V2R
V2O
V1N
V1R
V1O
PN
N2
N1
0.0
0.2
0.4
0.6
0.8
1.0
Activation

(a)

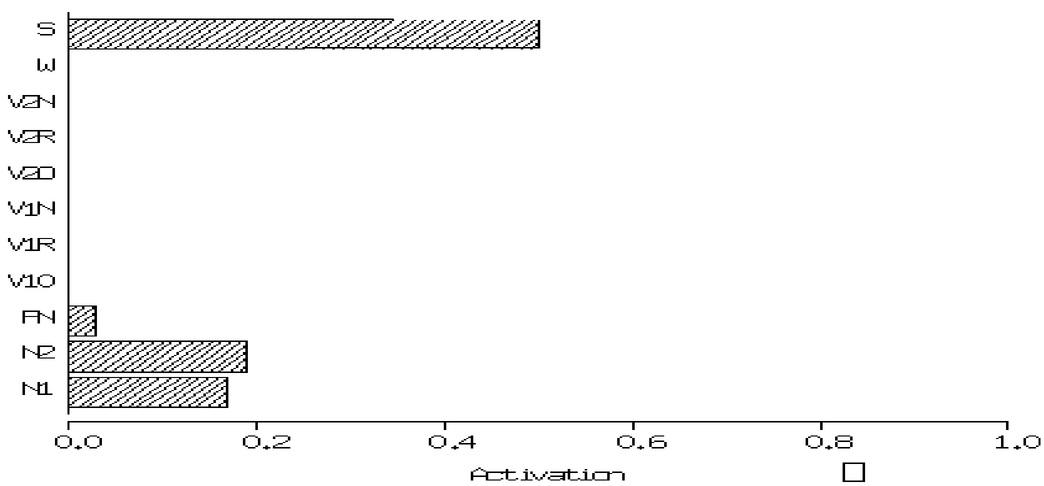

boys ...
S
W
V2N
V2R
V2O
V1N
V1R
V1O
PN
N2
N1
0.0
0.2
0.4
0.6
0.8
1.0
Activation

(b)

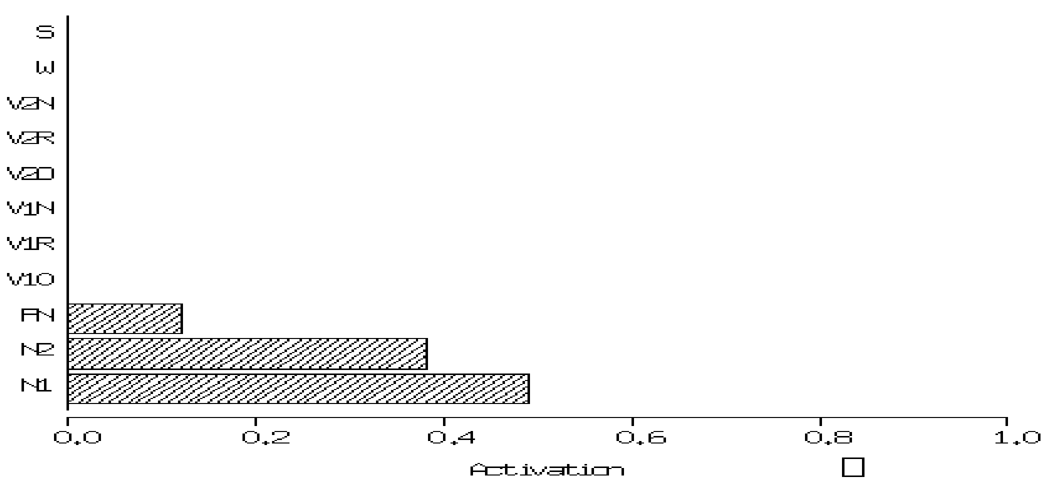
boy lives ...

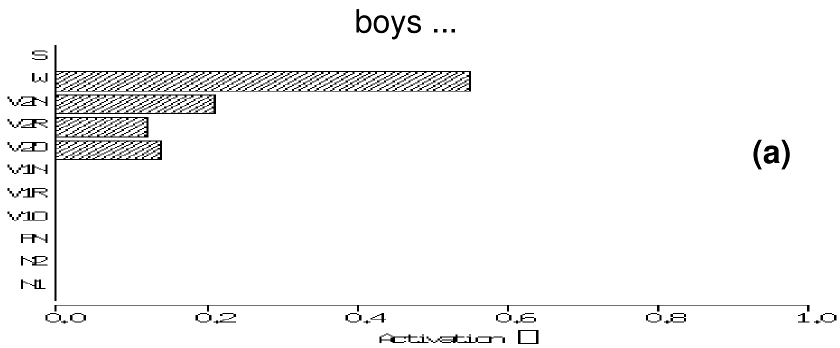


boy sees ...



boy chases ...



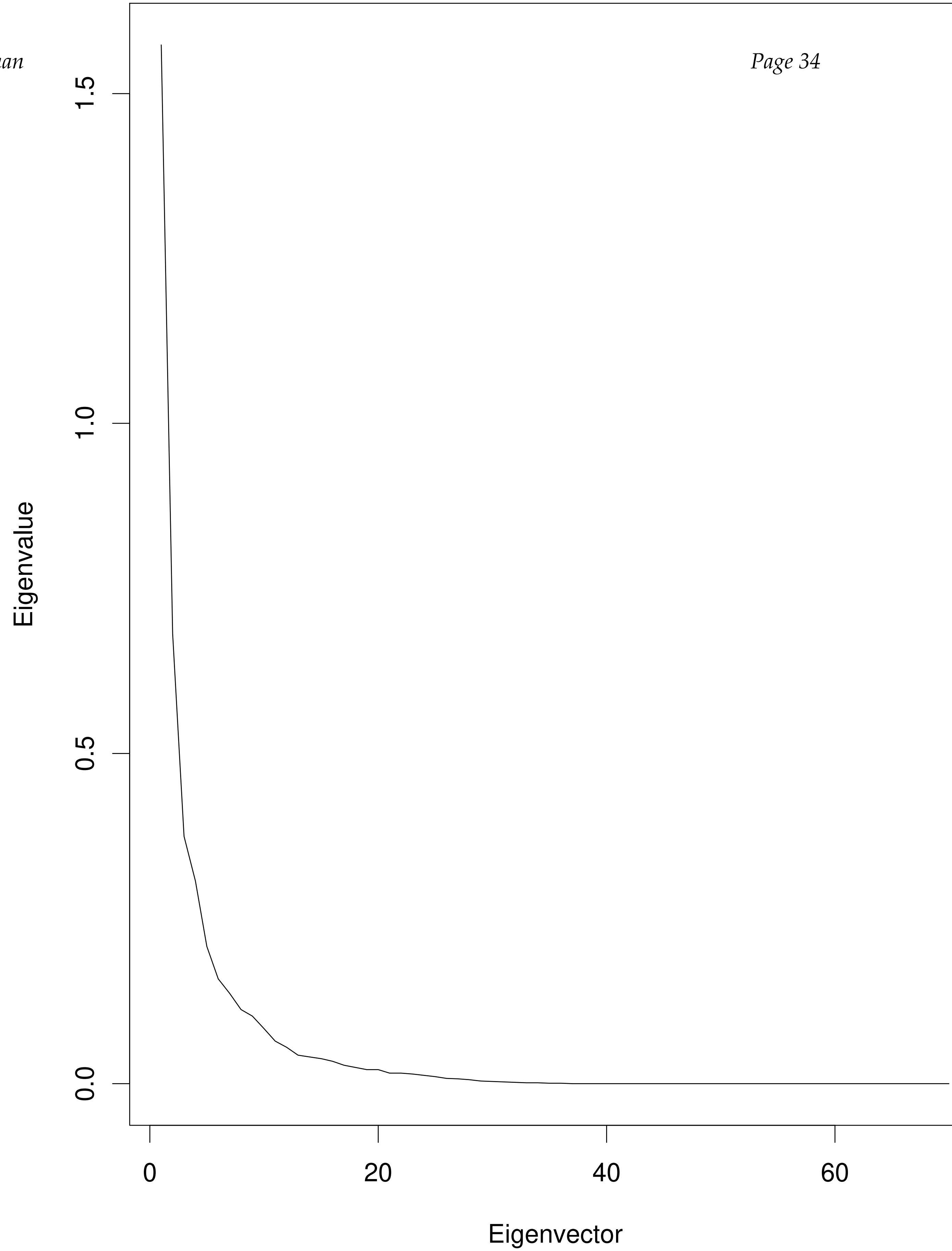

boys ...
S
W
V2N
V2R
V2D
V1N
V1R
V1D
PN
N2
N1
0.0
0.2
0.4
0.6
0.8
1.0
Activation

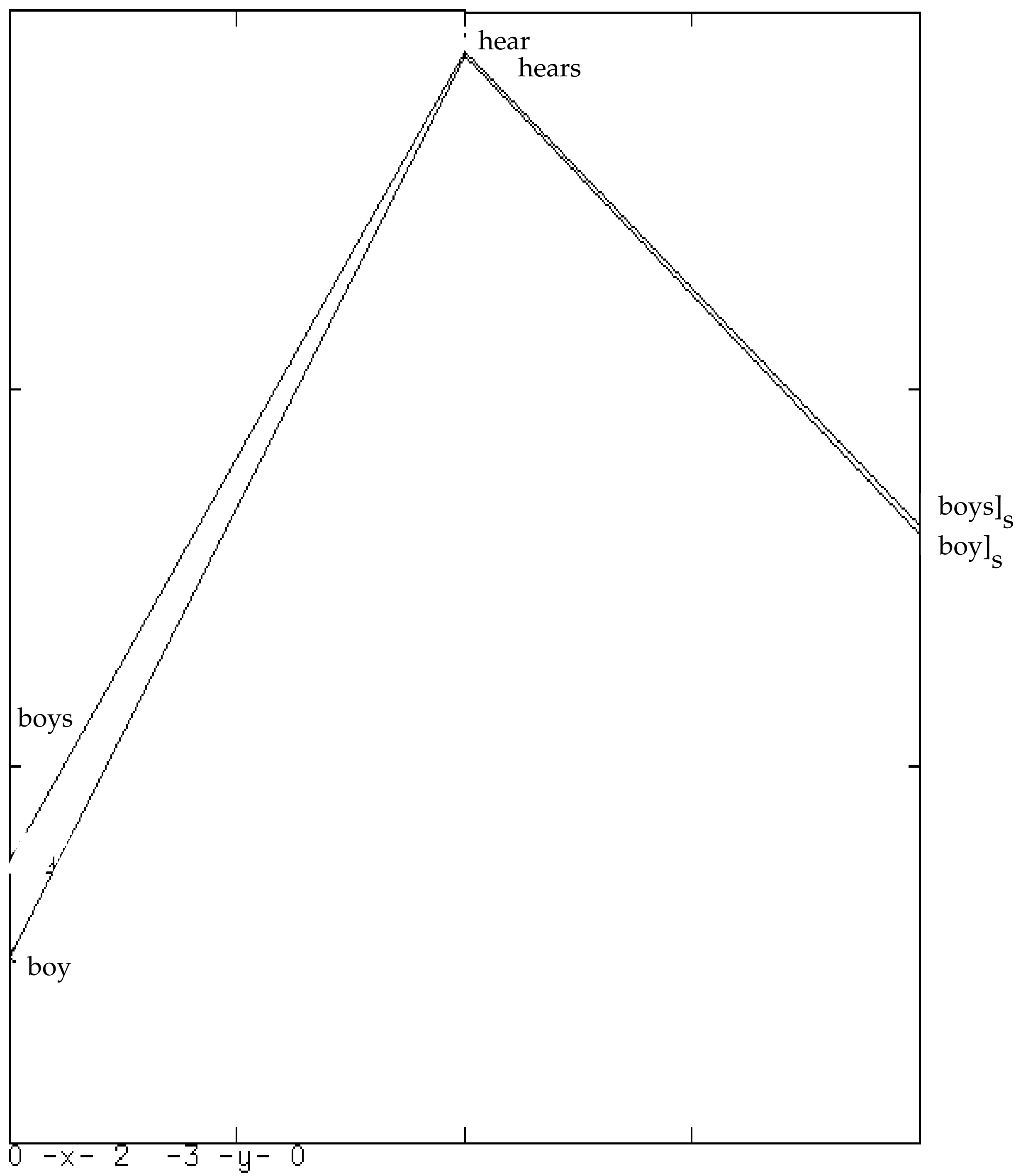
(a)

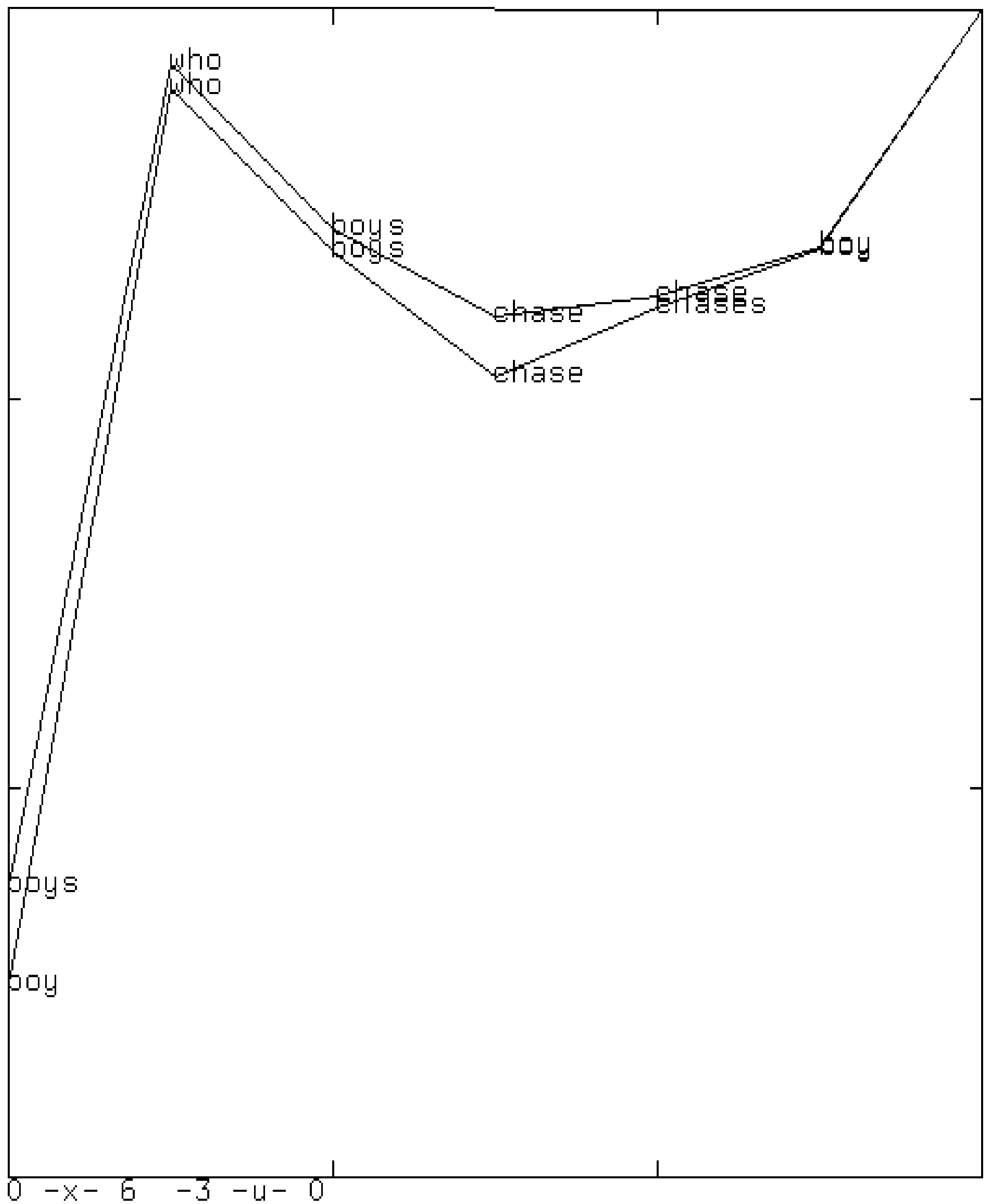

boys who ...
S
W
V2N
V2R
V2D
V1N
V1R
V1D
PN
N2
N1
0.0
0.2
0.4
0.6
0.8
1.0
Activation

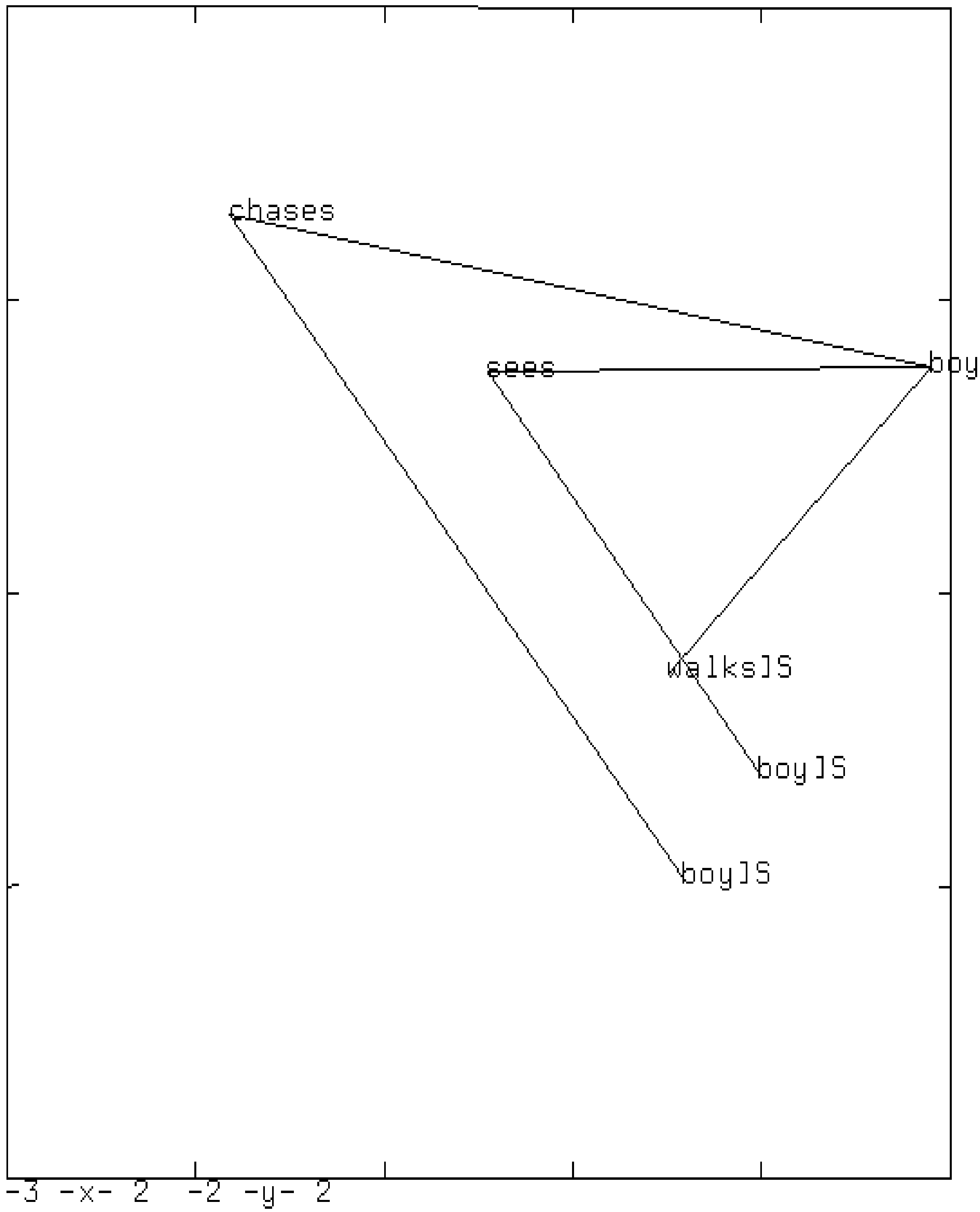
(b)

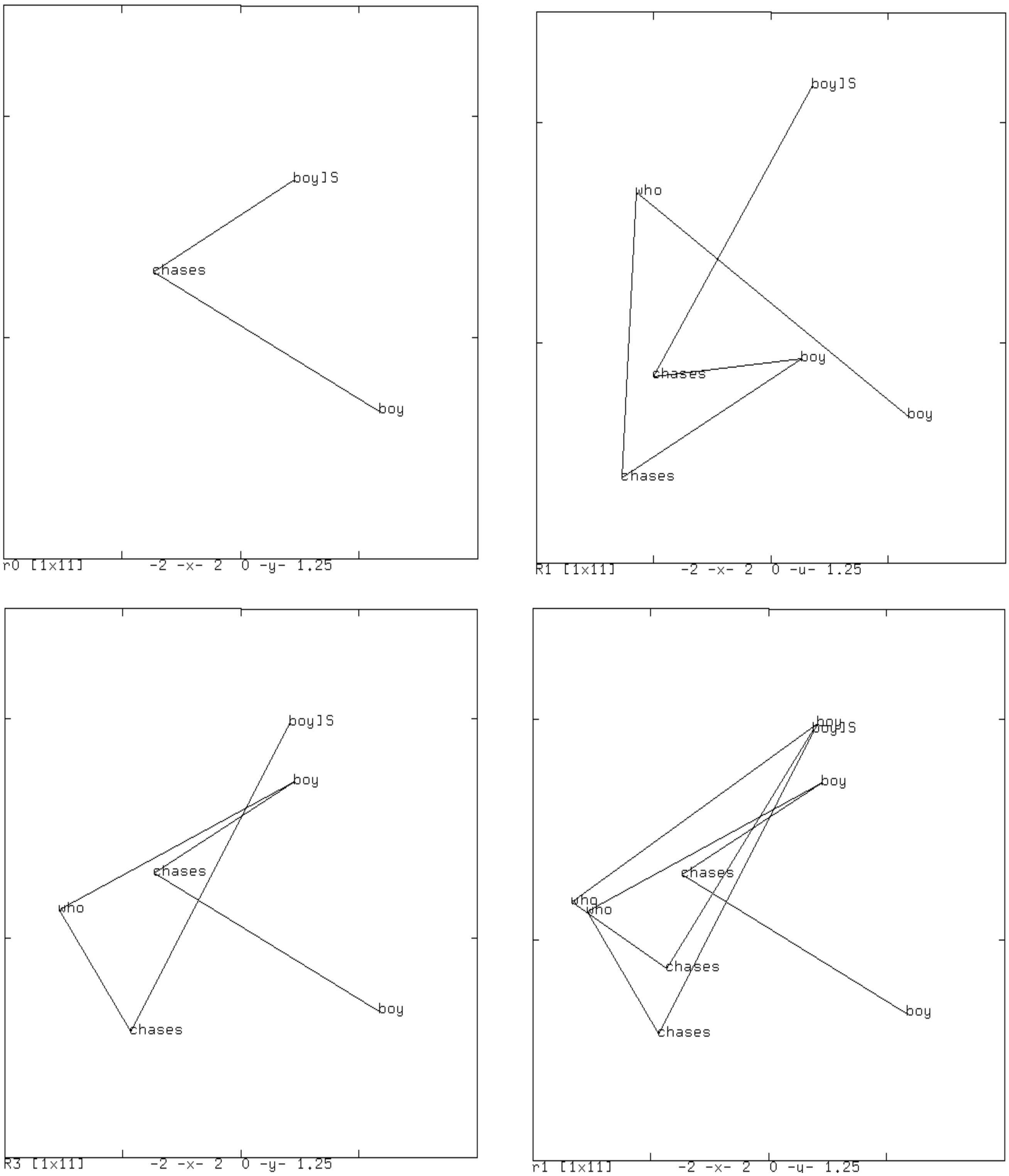

boys who mary ...
(c)
S
W
V2N
V2R
V2O
V1N
V1R
V1O
PN
N2
N1
0.0
0.2
0.4
0.6
0.8
1.0
Activation
boys who mary chases ...
(d)
Activation
boys who mary chases feed ...
(e)
Activation
boys who mary chases feed cats.
(f)
Activation


Eigenvalue
0.0
0.5
1.0
1.5
Eigenvector
0
20
40
60


hear
hears
boys]$_S$
boy]$_S$
boys
boy
0 -x- 2 -3 -y- 0


who
who
boys
boys
chase
chase
chase
chases
boy
boys
boy
0 -x- 6 -3 -u- 0


chases
sees
boy
walks]S
boy]S
boy]S
-3 -x- 2 -2 -y- 2


boy]S
chases
boy
r0 [1x11] -2 -x- 2 0 -y- 1.25
boy]S
who
chases
boy
boy
chases
R1 [1x11] -2 -x- 2 0 -u- 1.25
boy]S
boy
chases
who
boy
chases
R3 [1x11] -2 -x- 2 0 -y- 1.25
boy
boy]S
boy
chases
who
who
chases
boy
chases
r1 [1x11] -2 -x- 2 0 -u- 1.25

‘

S $\rightarrow$ **NP VP** "."
NP $\rightarrow$ **PropN** | **N** | **N RC**
VP $\rightarrow$ **V** (**NP**)
RC $\rightarrow$ *who* **NP VP** | *who* **VP** (**NP**)
N $\rightarrow$ *boy* | *girl* | *cat* | *dog* | *boys* | *girls* | *cats* | *dogs*
PropN $\rightarrow$ *John* | *Mary*
V $\rightarrow$ *chase* | *feed* | *see* | *hear* | *walk* | *live* | *chases* |
feeds | *sees* | *hears* | *walks* | *lives*

Additional restrictions:

- **number agreement between N & V within clause, and (where appropriate) between head N & subordinate V**

- **verb arguments:**
  hit, feed $\rightarrow$ **require a direct object**
  see, hear $\rightarrow$ **optionally allow a direct object**
  walk, live $\rightarrow$ **preclude a direct object**
  (observed also for head/verb relations in relative clauses)

Table 1