Language as a dynamical system

Jeffrey L. Elman
University of California, San Diego

1 Introduction

Despite considerable diversity among theories about how humans process language, there are a number of fundamental assumptions which are shared by most such theories. This consensus extends to the very basic question about what counts as a cognitive process. So although many cognitive scientists are fond of referring to the brain as a 'mental organ' (e.g., Chomsky, 1975)—implying a similarity to other organs such as the liver or kidneys—it is also assumed that the brain is an organ with special properties which set it apart. Brains 'carry out computation' (it is argued); they 'entertain propositions'; and they 'support representations'. Brains may be organs, but they are very different than the other organs found in the body.

Obviously, there are substantial differences between brains and kidneys, just as there are between kidneys and hearts and the skin. It would be silly to minimze these differences. On the other hand, a cautionary note is also in order. The domains over which the various organs operate are quite different, but their common biological substrate is quite similar. The brain is indeed quite remarkable, and does some things which are very similar to human-made symbol processors; but there are also profound differences between the brain and digital symbol processors, and attempts to ignore these on grounds of simplification or abstraction run the risk of fundamentally misunderstanding the nature of neural computation (Churchland & Sejnowski, 1992). In a larger sense, I raise the more general warning that (as Ed Hutchins has suggested) “cognition may not be what we think it is”. Among other things, I will suggest in this chapter that language (and cognition in general) may be more usefully understood as the behavior of a dynamical system. I believe this is a view which both acknowledges the similarity of the brain to other bodily organs and respects the evolutionary history of the nervous system, while also acknowledging the very remarkable properties possessed by the brain.

In the view I will outline, representations are not abstract symbols but rather regions of state space. Rules are not operations on symbols but rather embedded in the dynamics of the system, a dynamics which permits movement from certain regions to others while making other transitions difficult. Let me emphasize from the beginning that I am not arguing that language behavior is not rule-governed. Instead, I suggest that the *nature of the rules* may be different than what we have conceived them to be.

The remainder of this chapter is organized as follows. In order to make clear how the dynamical approach (instantiated concretely here as a connectionist network) differs from the standard approach, I begin by summarizing some of the central characteristics of the traditional approach to language processing. Then I shall describe a connectionist model which embodies different operating principles from the classical approach to symbolic computation. The results of several simulations using that architecture are presented and discussed. Finally, I will discuss some of the results which may be yielded by this perspective.

2 Grammar and the lexicon: The traditional approach

Language processing is traditionally assumed to involve a *lexicon*, which is the repository of facts concerning individual words, and a set of *rules* which constrain the ways those words can be combined to form sentences. From the point of view of a listener attempting to process spoken language, the initial problem involves taking acoustic input and retrieving the relevant word from the lexicon. This process is often supposed to involve separate stages of *lexical access* (in which contact is made with candidate words based on partial information), and *lexical recognition* (or retrieval, or selection; in which a choice is made on a specific word), although finer-grained distinctions may also be useful (e.g., Tyler & Frauenfelder, 1987). Subsequent to recognition, the retrieved word must be inserted into a data structure which will eventually correspond to a sentence; this procedure is assumed to involve the application of rules.

As described, this scenario may seem simple, straightforward, and not likely to be controversial. But in fact, there is considerable debate about a number of important details. For instance:

Is the lexicon passive or active? In some models, the lexicon is a passive data structure (Forster, 1976). In other models, lexical items are active (Marslen-Wilson, 1980; McClelland & Elman, 1986; Morton, 1979) in the style of Selfridge's "demons" (Selfridge, 1958).

How is the lexicon organized and what are its entry points? In active models, the internal organization of the lexicon is less an issue, because the lexicon is also usually content addressable, so that there is direct and simultaneous contact between an unknown input and all relevant lexical representations. With passive models, an additional look-up process is required and so the organization of the lexicon becomes more important for efficient and rapid search. The lexicon may be organized along dimensions which reflect phonological, or orthographic, or syntactic, or syntactic properties; or it may be organized along usage parameters, such as frequency (Forster, 1976). Other problems include how to catalog morphologically related elements (e.g., are "telephone" and "telephonic" separate entries? "girl" and "girls"? "ox" and "oxen"?); how to represent words with multiple meanings (the various meanings of "bank" may be different enough to warrant distinct entries, but what about the various meanings of “run”, some of which are only subtly different, and others which have more distant but still clearly related meanings?); whether the lexicon includes information about argument structure; and so on.

Is recognition all-or-nothing, or graded? In some theories, recognition occurs at the point where a spoken word becomes uniquely distinguished from its competitors (Marslen-Wilson, 1980). In other models, there may be no consistent point where recognition occurs; rather, recognition is a graded process subject to interactions which may hasten or slow down the retrieval of a word in a given context. The recognition point is a strategically-controlled threshold (McClelland & Elman, 1986).

How do lexical competitors interact? If the lexicon is active, there is the potential for interactions between lexical competitors. Some models build inhibitory interactions between words (McClelland & Elman, 1986); others have suggested that the empirical evidence rules out word-word inhibitions (Marslen-Wilson, 1987).

How are sentence structures constructed from words? This single question has given rise to a vast and complex literature. The nature of the sentence structures themselves are fiercely debated, reflecting the diversity of current syntactic theories. There is in addition considerable controversy around the sort of information which may play a role in the construction process, or the degree to which at least a first-pass parse is restricted to the purely syntactic information available to it (Frazier & Rayner, 1982; Trueswell, Tanenhaus, & Kello, 1992).

There are thus a considerable number of questions which remain open. Nonetheless, I believe it is accurate to say that there is also considerable consensus regarding certain fundamental principles. I take this consensus to include the following.

(a) A commitment to *discrete* and *context-free symbols*. This is more readily obvious in the case of the classical approaches, but many connectionist models utilize localist representations in which entities are discrete and atomic (although graded activations may be used to reflect uncertain hypotheses).

A central feature of all of these forms of representation—localist connectionist as well as symbolic—is that they are *intrinsically context-free*. The symbol for a word, for example, is the same regardless of its usage. This gives such systems great combinatorial power, but it also limits their ability to reflect idiosyncratic or contextually-specific behaviors.

This assumption also leads to a distinction between *types* and *tokens* and motivates the need for *variable binding*. Types are the canonical context-free versions of symbols; tokens are the versions which are associated with specific contexts; and binding is the operation which enforces the association (e.g., by means of indices, subscripts, or other diacritics).

(b) The view of *rules as operators* and the *lexicon as operands*. Words in most models are conceived of as the objects of processing. Even in models in which lexical entries may be active, once word a is recognized it becomes subject to grammatical rules which build up higher-level structures.

(c) The *static nature of representations*. Although the processing of language clearly unfolds over over time, the representations which are produced by traditional models typically have a curiously static quality. This is revealed in several ways. For instance, it is assumed that the lexicon pre-exists as a data structure in much the same way that a dictionary exists independently of its use. Similarly, the higher-level structures created during sentence comprehension are built up through an accretive process, and the successful product of comprehension will be a mental structure in which all the constituent parts (words, categories, relational information) are simultaneously present. (Presumably these become inputs to some subsequent interpretive process which constructs discourse structures.) That is, although processing models ("performance models") often take seriously the temporal dynamics involved in computing target structures, the target structures themselves are inherited from theories which ignore temporal considerations (“competence models”).

(d) The *building metaphor*. In the traditional view, the act of constructing mental representations is similar to the act of constructing a physical edifice. Indeed, this is precisely what is claimed in the Physical Symbol System Hypothesis (Simon, 1980). In this view, words and more abstract constituents are like the bricks in a building; rules are the mortar which binds them together. As processing proceeds, the representation grows much as does a building under construction. Successful processing results in a mental edifice which is a complete and consistent structure, again, much like a building.

I take these assumptions to be widely shared among researchers in the field of language processing, although they are rarely stated explicitly. Furthermore, these assumptions have formed the basis for a large body of empirical literature; they have played a role in the framing of the questions which are posed, and later in interpreting the experimental results. Certainly it is incumbent on any theory which is offered as replacement to at least provide the framework for describing the empirical phenomena, as well as improving our understanding of the data.

Why might we be interested in another theory? One reason is that this view of our mental life which I have just described, that is, a view which relies on discrete, static, passive, and context-free representations, appears to be sharply at variance with what is known about the computational properties of the brain (Churchland & Sejnowski, 1992). It must also be acknowledged that while the theories of language which subscribe to the assumptions listed above do provide a great deal of coverage of data, that coverage is often flawed, internally inconsistent and *ad hoc*, and highly controversial. So it is not unreasonable to raise the question: Do the shortcomings of the theories arise from assumptions which are basically flawed? Might there be other, better ways of understanding the nature of the mental processes and representations which underlie language? In the next section, I would like to suggest an alternative view of computation, in which language processing is seen as taking place in a dynamical system. The lexicon is viewed as consisting of regions of state space within that system; the grammar consists of the dynamics (attractors and repellers) which constrain movement in that space. As we will see, this approach entails representations which are highly context-sensitive, continuously varied and probabilistic (but of course 0.0 and 1.0 are also probabilities), and in which the objects of mental representation are better thought of as trajectories through mental space rather than things which are constructed.

An entry-point to describing this approach is the question of how one deals with time and the problem of serial processing. Language, like many other behaviors, unfolds and is processed over time. This simple fact—so simple it seems trivial—turns out to be problematic when explored in detail. Therefore, I turn now to the question of time. I describe a connectionist approach to temporal processing and show how it can be applied to several linguistic phenomena. In the final section I turn to the pay-off and attempt to show how this approach leads to useful new views about the lexicon and about grammar.

3 The problem of time

Time is the medium in which all our behaviors unfold; it is the context within which we understand the world. We recognize *causality* because causes precede effects; we learn that coherent motion over time of points on the retinal array is a good indicator of *objecthood*; and it is difficult to think about phenomena such as *language*, or *goal-directed behavior*, or *planning* without some way of representing time. Time’s arrow is such a central feature of our world that it is easy to think that, having acknowledged its pervasive presence, little more needs to be said.

But time has been the stumbling block of many theories. An important issue in models of motor activity, for example, has been the nature of the motor intention. Does the action plan consist of a literal specification of output sequences (probably not), or does it represent serial order in a more abstract manner (probably so, but how; e.g., Fowler, 1977; Jordan & Rosenbaum, 1988; Kelso, Saltzman, & Tuller, 1986; MacNeilage, 1970)? Within the realm of natural language processing, there is considerable controversy about how information accumulates over time and what information is available when (e.g., Altmann & Steedman, 1988; Ferreira & Henderson, 1990; Trueswell, Tanenhaus, & Kello, in press).

Time has been a challenge for connectionist models as well. Early models, perhaps reflecting the initial emphasis on the parallel aspects of these models, typically adopted a spatial representation of time (e.g., McClelland & Rumelhart, 1981). The basic approach is illustrated in Figure 1. The temporal order of input events (first-to-last) is represented by the spatial order (left-to-right) of the input vector. There are a number of problems with this approach (see Elman, 1990, for discussion). One of the most serious is that the left-to-right spatial ordering has no intrinsic significance at the level of computation which is meaningful for the network. All input dimensions are orthogonal to each other in the input vector space. The human eye tends to see patterns such as 01110000 and 00001110 as having undergone a spatial (or temporal, if we understand these as representing an ordered sequence) translation, because the notation suggests a special relationship may exist between adjacent bits. But this relationship is the result of considerable processing by the human visual system, and is not intrinsic to the vectors themselves. The first element in a vector is not “closer” in any useful sense to the second element than it is to the last element. Most important, is not available to simple networks of the form shown in Figure 1. A particularly unfortunate consequence is that there is no basis in such architectures for generalizing what has been learned about spatial or temporal stimuli to novel patterns.

—Insert Figure 1—

More recent models have explored what is intuitively a more appropriate idea: Let time be represented by the effects it has on processing. If network connections include feedback loops, then this goal is achieved naturally. The state of the network will be some function of the current inputs plus the network's prior state. Various algorithms and architectures have been developed which exploit this insight (e.g., Elman, 1990; Jordan, 1986; Mozer, 1989; Pearlmutter, 1989; Rumelhart, Hinton, & Williams, 1986). Figure 2 shows one architecture, the Simple Recurrent Network, which was used for the studies to be reported here.

—Insert Figure 2—

In the SRN architecture, at time *t* hidden units receive external input, and also collateral input from themselves at time *t-1* (the context units are simply used to implement this delay). The activation function for any given hidden unit h_i is the familiar logistic,

$$f(h_i) = \frac{1}{1 + e^{-net}}$$

but where the net input to the unit at time *t*, $net_i(t)$, is now

$$net_i(t) = \sum_j w_{ij} a_j(t) + b_i + \sum_k w_{ik} h_k(t-1)$$

That is, the net input on any given tick of the clock *t* includes not only the weighted sum of inputs and the node's bias, but the weighted sum of the hidden unit vector at the prior time step. (Henceforth, when referring to the state space of this system, I shall be referring specifically to the *k*-dimensional space defined by the *k* hidden units.)

In the typical feedforward network, hidden units develop representations which enable the network to perform the task at hand (Rumelhart, Hinton, &

Williams, 1986). These representations may be highly abstract and are function-based. That is, the similarity structure of the internal representations reflects the demands of the task being learned, rather than the similarity of the inputs' form. When recurrence is added, the hidden units assume an additional function. They now provide the network with memory. But as is true in the feedforward network, the encoding of the temporal history is task-relevant and may be highly abstract; it rarely is the case that the encoding resembles a verbatim tape-recording.

One task for which the SRN has proven useful is prediction. There are several reasons why it is attractive to train a network to predict the future. One which arises with supervised learning algorithms such as backpropagation of error is the question of where the teaching information comes from. In many cases, there are plausible rationales which justify the teacher. But the teacher also reflects important theoretical biases which one might sometimes like to avoid (for example, if one were interested in using the network to generate alternative theories). Since the teacher in the prediction task is simply the time-lagged input, it represents information which is directly observable from the environment and is relatively theory neutral. Furthermore, there is good reason to believe that anticipating the future plays an important role in learning about the world. Finally, prediction is a powerful tool for learning about temporal structure. Insofar as the order of events may reflect upon the past in complex and non-obvious ways, the network will be required to develop relatively abstract encodings of these dependencies in order to generate successful predictions.

The SRN architecture, as well as other forms of recurrent networks, have been used in a variety of applications and has yielded promising results. The SRN's ability to handle temporal sequences makes it a particularly relevant architecture for modeling language behaviors. The deeper question which then arises is whether the solutions found by such recurrent network architectures differ in any substantial ways from more traditional models. And if the solutions are different, are these differences positive or negative?

4 Rules and representations: A dynamical perspective

We begin with the observation that networks such as that in Figure 2 are dynamical systems. This means that their state at any given point in time is some function which reflects their prior state (see Norton, this volume, for a detailed review of the definition and characteristics of dynamical systems). The computational properties of such networks are not yet fully known, but it is clear that they are considerable (Siegelmann & Sontag, 1992). It also seems reasonable that the conceptual notions which are associated with discrete automata theory and symbolic computation may offer less insight into their functioning than the concepts from dynamical systems theory (e.g., Pollack, 1990). How might such networks be applied to problems relevant to language processing, and how might they suggest a different view of the underlying mechanisms of language? One way to approach this is to consider the problem of how the elements of language may be ordered.

Language is a domain in which the ordering of elements is particularly complex. Word order, for instance, reflects the interaction of multiple factors. These include syntactic constraints, semantic and pragmatic goals, discourse considerations, and processing constraints (e.g., verb-particle constructions such as “run up” may be split by a direct object, but not when the noun phrase is long enough to disrupt the processing of the discontinuous verb as a unit). Whether or not one subscribes to the view that these knowledge sources exert their effects autonomously or interactively, there is no question that the final output—the word stream—reflects their joint interplay.

We know also that the linear order of linguistic elements provides a poor basis for characterizing the regularities which exist within a sentence. A noun may agree for number with a verb which immediately follows it (as in 1a) or which is separated by an arbitrarily great distance (as in 1b):

1. (a) The **children**$_{pl}$ **like**$_{pl}$ ice cream.

   (b) The **girl**$_{sg}$ who Emily baby-sits for every other Wednesday while her parents go to nightschool **likes**$_{sg}$ ice cream.

Such considerations led Miller and Chomsky (1963) to argue that statistically-based algorithms are infeasible for language learning, since the number of sentences which a listener would need to hear in order to know precisely which of the 14 words which precede *likes* in (1b) determines the correct number for *likes* would vastly outnumber the data available (in fact, even conservative estimates suggest that more time would be needed than is available in an entire individual's lifetime). On the other hand, recognition that the dependencies respect an underlying hierarchical structure vastly simplifies the problem: Subject nouns in English agree for number with their verbs; embedded clauses may intervene but do not participate in the agreement process.

One way to challenge a simple recurrent network with a problem which has some relevance to language would therefore be to attempt to train it to predict the successive words in sentences. We know that this is a hard problem which cannot be solved in any general way by simple recourse to linear order. We know also that this is a task which has some psychological validity. Human listeners are able to predict word endings from beginnings; listeners can predict grammaticality from partial sentence input; and sequences of words which violate expectations—i.e., which are unpredictable—result in distinctive electrical activity in the brain. An interesting question is whether a network could be trained to predict successive words. In the following two simulations we shall see how, in the course of solving this task, the network develops novel representations of the lexicon and of grammatical rules.

4.1 The lexicon as structured state space

Words may be categorized with respect to many factors. These include such traditional notions as *noun, verb,* etc.; the argument structures they are associated with; and semantic features. Many of these characteristics are predictive of a word's syntagmatic properties. But is the reverse true? Can distributional facts be used to infer something about a word's semantic or categorial features? The goal of the first simulation was to see if a network could work backwards in just this sense.

A small lexicon of 29 nouns and verbs was used to form simple sentences (see Elman, 1990, for details). Each word was represented as a localist vector in which a single randomly assigned bit was turned on. This input representation ensured that there was nothing about the form of the word which was correlated with its properties, and thus that any classifications would have to be discovered by the network based solely on distributional behavior.

A network similar to the one shown in Figure 2 was trained on a set of 10,000 sentences, with each word presented in sequence to the network and each sentence concatenated to the preceding sentence. The task of the network was to predict the successive word. After each word was input, the output (which was the prediction of the next input) was compared with the actual next word and weights were adjusted by the backpropagation of error learning algorithm.

At the conclusion of training, the network was tested by comparing its predictions against the corpus. Since the corpus was non-deterministic, it was not reasonable to expect that the network (short of memorizing the sequence) would be able to make exact predictions. Instead, the network predicted the cohort of *potential* word successors in each context. The activation of each cohort turned out to be highly correlated with the conditional probability of each word, in that context (the mean cosine of the output vector with the empirically derived probability distribution was 0.916).

This behavior suggests that in order to maximize performance at prediction, the network identifies inputs as belonging to classes of words based on distributional properties and co-occurrence information. These classes were not represented in the overt form of the word, since these were all orthogonal to each other. However, the network is free to learn internal representations at the hidden unit layer which might capture this implicit information.

To test this possibility, the corpus of sentences was run through the network a final time. As each word was input, the hidden unit activation pattern which was produced by the word, plus the context layer, was saved. For each of the 29 words, a mean vector was computed, averaging across all instances of the word in all contexts. These mean vectors were taken to be prototypes, and were subjected to hierarchical clustering. The point of this was to see whether the inter-vector distances revealed anything about similarity structure of the hidden unit representation space (Euclidean distance being taken as a measure of similarity). The tree in Figure 3 was then constructed from that hierarchical clustering.

—Insert Figure 3 about here—

The similarity structure revealed in this tree indicates that the network discovered several major categories of words. The two largest categories correspond to the input vectors which are verbs and nouns. The verb category is subdivided into those verbs which require a direct object, those which are intransitive, and those for which (in this corpus) a direct object was optional. The noun category is broken into animates and inanimates. The animates contain two classes: human and nonhuman, with nonhumans are subdivided into large animals and small animals. The inanimates are divided into breakables, edibles, and miscellaneous.

First, it must be said that the network obviously knows nothing about the real semantic content of these categories. It has simply inferred that such a category structure exists. The structure is inferred because it provides the best basis for accounting for distributional properties. Obviously, a full account of language would require an explanation of how this structure is given content (grounded in the body and in the world). But it is interesting that the evidence for the structure can be inferred so easily on the basis only of form-internal evidence, and this result may encourage caution about just how much information is implicit in the data and how difficult it may be to use this information to construct a framework for conceptual representation.

However, my main point is not to suggest that this is the primary way in which grammatical categories are acquired by children, although I believe that cooccurrence information may indeed play a role in such learning. The primary thing I would like to focus on is what this simulation suggests about the nature of representation in systems of this sort. That is, I would like to consider the representational properties of such networks, apart from the specific conditions which give rise to those representations.

Where is the lexicon in this network? Recall the earlier assumptions: The lexicon is typically conceived of as a passive data structure. Words are objects of processing. They are first subject to acoustic/phonetic analysis, and then their internal representations must be accessed, recognized, and retrieved from permanent storage. Following this, the internal representations have to be inserted into a grammatical structure.

The status of words in a system of the sort described here is very different: Words are not the *objects* of processing as much as they are inputs which *drive* the processor in a more direct manner. As Wiles and Bloesch (1992) have suggest, it is more useful to understand inputs to networks of this sort as *operators* rather than as *operands.* Inputs operate on the network's internal state and move it to another position in state space. What the network learns over time is what response it should make to different words, taking context into account. Because words have reliable and systematic effects on behavior, it is not surprising that all instances of a given word should result in states which are tightly clustered, or that grammatically or semantically related words should produce similar effects on the network. We might choose to think of the internal state that the network is in when it processes a word as representing that word (in context), but it is more accurate to think of that state as the *result* of processing the word, rather than as a representation of the word itself.

Note that there is an implicitly hierarchical organization to the regions of state space associated with different words. This organization is achieved through the spatial structure. Conceptual similarity is realized through position in state space. Words which are conceptually distant produce hidden unit activation patterns which are spatially far apart. Higher-level categories correspond to large regions of space; lower-level categories correspond to more restricted subregions. For example, *dragon* is a noun and causes the network to move into the noun region of the state space. It is also [+animate], which is reflected in the subregion of noun space which results. Because large animals typically are described in different terms and do different things than small animals, the general region of space corresponding to *dragon, monster* and *lion* is distinct from that occupied by *mouse, cat,* and *dog.* The boundaries between these regions may be thought of as hard in some cases (e.g., nouns are very far from verbs) or soft in others (e.g., *sandwich*, *cookie,* and *bread* are not very far from *car*, *book*, and *rock*. One even might imagine cases where in certain contexts, tokens of one word might overlap with tokens of another. In such cases, one would say that the system has generated highly similar construals of the different words.

4.2 Rules as attractors

If the lexicon is represented as regions of state space, what about rules? We have already seen that some aspects of grammar are captured in the tokenization of words, but this is a fairly limited sense of grammar. The well-formedness of sentences depends on relationships which are not readily stated in terms of simple linear order. Thus the proper generalization about why the main verb in (1b) is in the plural is that the main subject is plural, and not that the word 14 words prior was a plural noun. The ability to express such generalizations would seem to require a mechanism for explicitly representing abstract grammatical structure, including constituent relationships (e.g., the notion that some elements are part of others). Notations such as phrase structure trees (among others) provide precisely this capability. It is not obvious how complex grammatical relations might be expressed using distributed representations. Indeed, it has been argued that distributed representations (of the sort exemplified by the hidden unit activation patterns in the previous simulation) cannot have constituent structure in any systematic fashion (Fodor & Pylyshyn, 1988). (As a backup, Fodor and Pylyshyn suggest that if distributed representations *do* have a systematic constituent structure, then they are merely implementations of what they call the “classical” theory, in this case, the Language of Thought, Fodor, 1976.)

The fact that the grammar of the first simulation was extremely simple made it difficult to explore these issues. Sentences were all declarative and monoclausal. This simulation sheds little light on the grammatical potential of such networks.

A better test would be train the network to predict words in complex sentences which contain long-distance dependencies. This was done in Elman (1991b) using a strategy which was similar to the one outlined in the prior simulation, except that sentences had the following characteristics:

(1) Nouns and verbs agreed for number. Singular nouns required singular verbs; plural nouns selected plural verbs.

(2) Verbs differed with regard to their verb argument structure. Some verbs were transitive; others were intransitive; and others were optionally transitive.

(3) Nouns could be modified by relative clauses. Relative clauses could either be object-relatives (the head had the object role in the clause) or subject-relative (the head was the subject of the clause), and either subject or object nouns could be relativized.

As in the previous simulation, words were represented in localist fashion so that information about neither the grammatical category (noun or verb) nor the number (singular or plural) was contained in the form of the word. The network also only saw positive instances; only grammatical sentences were presented.

The three properties interact in ways which were designed to make the prediction task difficult. The prediction of number is easy in a sentence such as (2a), but harder in (2b).

2. (a) The boys$_{pl}$ chase$_{pl}$ the dogs.

(b) The boys$_{pl}$ who the dog$_{sg}$ chases$_{sg}$ run$_{pl}$ away.

In the first case, the verb follows immediately. In the second case, the first noun agrees with the second verb (*run)* and is plural; the verb which is actually closest to it (*chase*) is in the singular because it agrees with the intervening word (*dog*).

Relative clauses cause similar complications for verb argument structure. In (3), it is not difficult for the network to learn that *chase* requires a direct object, *see* permits (but does not require) one, and *lives* is intransitive.

3. (a) The cats chase the dog.

(b) The girls see. The girls see the car.

(c) The patient lives.

On the other hand, consider (4):

4. The dog who the cats chase run away.

The direct object of the verb *chase* in the relative clause is *dog.* However, *dog* is also the head of the clause (as well as the subject of the main clause). *Chase* in this grammar is obligatorily transitive, but the network must learn that when it occurs in such structures the object position is left empty (gapped) because the direct object has already been mentioned (filled) as the clause head.

These data illustrate the sorts of phenomena which have been used by linguists to argue for abstract representations with constituent structure (Chomsky, 1957); they have also been used to motivate the claim that language processing requires some form of pushdown store or stack mechanism. They therefore impose a difficult set of demands on a recurrent network.

However, after training a network on such stimuli (Elman, 1991b) it appeared the network was able to make correct predictions (mean cosine between outputs and empirically derived conditional probability distributions: 0.852; perfect performance would have been 1.0). These predictions honored the grammatical constraints which were present in the training data. The network was able to correctly predict the number of a main sentence verb even in the presence of intervening clauses (which might have the same or conflicting number agreement between nouns and verbs). The network also not only learned about verb argument structure differences, but correctly “remembered” when an object-relative head had appeared, so that it would not predict a noun following an embedded transitive verb. Figure 4 shows the predictions made by the network during testing with a novel sentence.

—Insert Figure 4 about here—

How is this behavior achieved? What is the nature of the underlying knowledge possessed by the network which allows it to perform in a way which conforms with the grammar? It is not likely that the network simply memorized the training data, because the network was able to generalize its performance to novel sentences and structures it had never seen before. But just how general was the solution, and just how systematic?

In the previous simulation, hierarchical clustering was used to measure the similarity structure between internal representations of words. This gives us an indirect means of determining the spatial structure of the representation space. It does not let us actually determine what that structure is.[1] So one would like to be able to visualize the internal state space more directly. This is also important because it would allow us to study the ways in which the network's internal state changes over time as it processes a sentence. These trajectories might tell us something about how the grammar is encoded.

One difficulty which arises in trying to visualize movement in the hidden

[1] For example, imagine a tree with two major branches, each of which has two sub-branches. We can be certain that the items on the major branches occupy different regions of state space. More precisely, they lie along the dimension of major variation in that space. We can say nothing about other dimensions of variation, however. Two sub-branches may divide in similar ways. For example, [+human] and [-human] animates may each have branches for [+large] and [-large] elements. But it is impossible to know whether [+large, +human] and [+large, -human] elements differ from their [-large] counterparts in exactly the same way, i.e., by lying along some common axis corresponding to size. Clustering tells us only about distance relationships, not about the organization of space which underlies those relationships.

unit activation space over time is that it is an extremely high-dimensional space (70 dimensions, in the current simulation). These representations are distributed, which typically has the consequence that interpretable information cannot be obtained by examining activity of single hidden units. Information is more often encoded along dimensions which are represented across multiple hidden units.

This is not to say, however, that the information is not there, of course, simply that one needs to discover the proper viewing perspective to get at it. One way of doing this is to carry out a principal components analysis (PCA) over the hidden unit activation vectors. PCA allows us to discover the dimensions along which there is variation in the vectors; it also makes it possible to visualize the vectors in a coordinate system which is aligned with this variation. This new coordinate system has the effect of giving a somewhat more localized description to the hidden unit activation patterns. Since the dimensions are ordered with respect to amount of variance accounted for, we can now look at the trajectories of the hidden unit patterns along selected dimensions of the state space.[2]

—Insert Figures 5, 6, and 7 about here—

In Figures 5, 6, and 7 we see the movement over time through various plans in the hidden unit state space as the trained network processes various test sentences. Figure 5 compares the path through state space (along the second principal component) as the network processes the sentences *boys hear boys* and *boy hears boy*. PCA 2 encodes the number of the main clause subject noun, and

[2] There are limitations to the use of PCA to analyze hidden unit vectors. PCA yields a rotation of the original coordinate system which requires that the new axes be orthogonal. However, it need not be the case that the dimensions of variation in the hidden unit space are orthogonal to one another; this is especially true if the output units which receive the hidden unit vectors as input are nonlinear. It would be preferable to carry out a nonlinear PCA or use some other technique which both relaxed the requirement of orthogonality and which took into account the effect of the hidden-to-output nonlinearity.

the difference in the position along this dimension correlates with whether the subject is singular or plural. Figure 6 compares trajectories for sentences with verbs which have different argument expectations; *chases* requires a direct object, *sees* permits one, and *walks* precludes one. As can be seen, these differences in argument structure are reflected in a displacement in state space from upper left to lower right. Finally, Figure 7 illustrates the path through state space for various sentences which differ in degree of embedding. The actual degree of embedding is captured by the displacement in state space of the embedded clauses; sentences with multiple embeddings appear somewhat as spirals.

These trajectories illustrate the general principle at work in this network. The network has learned to represent differences in lexical items as different regions in the hidden unit state space. The sequential dependencies which exist among words in sentences are captured by the movement over time through this space as the network processes successive words in the sentence. These dependencies are actually encoded in the weights which map inputs (i.e., the current state plus new word) to the next state. The weights may be thought of as implementing the grammatical rules which allow well-formed sequences to be processed and to yield valid expectations about successive words. Furthermore, the rules are general. The network weights create attractors in the state space, so that the network is able to respond sensibly to novel inputs, as when unfamiliar words are encountered in familiar contexts.

5 Discussion

The image of language processing just outlined does not look very much like the traditional picture which we began with. Instead of a dictionary-like lexicon, we have a state space partitioned into various regions. Instead of symbolic rules and phrase structure trees, we have a dynamical system in which grammatical constructions are represented by trajectories through state space. Let me now consider what implications this approach might have for understanding several aspects of language processing.

5.1 Beyond sentences

Although I have focused here on processing of sentences, obviously language processing in real situations typically involves discourse which extends over many sentences. It is not clear, in the traditional scheme, how information which is represented in sentence structures might be kept available for discourse purposes. The problem is just that on the one hand, there are clearly limitations on how much information can be stored, so obviously not everything can be preserved; but on the other hand, there are many aspects of sentence-level processing which may be crucially affected by prior sentences. These include not only anaphora, but also such things as argument structure expectations (e.g., the verb *to give* normally requires a direct object and an indirect object, but in certain contexts these need not appear overtly if understood: *Do you plan to give money to the United Way? No, I gave last week.*).

The network's approach to language processing handles such requirements in a natural manner. The network is a system which might be characterized as highly opportunistic. It learns to perform a task, in this case prediction, doing just what it needs to do. Notice that in Figure 5, for example, the information about the number of the subject noun is maintained only until the verb which agrees with the subject has been processed. From that point on, the two sentences are identical. This happens because once the verb is encountered, subject number is no longer relevant to any aspect of the prediction task. (This emphasizes the importance of the task, because presumably tasks other than prediction could easily require that the subject number be maintained for longer.)

This approach to preserving information suggests that such networks would readily adapt to processing multiple sentences in discourse, since there is no particular reanalysis or re-representation of information which is required at sentence boundaries and no reason why some information cannot be preserved across sentences. Indeed, St. John (1992) and Harris & Elman (1990) have demonstrated that networks of this kind readily adapt to processing paragraphs and short stories. (The emphasis on functionality is reminiscent of suggestions made by Agre & Chapman (1987) and Brooks (1989). These authors argue that animals need not perfectly represent everything which is in their environment, nor store it indefinitely. Instead, they need merely be able to process that which is relevant to the task at hand.)

5.2 Types and tokens

Consider the first simulation, and the network's use of state space to represent words. This is directly relevant to the way in which the system addresses the *types*/*token* problem which arises in symbolic systems.

In symbolic systems, because representations are abstract and context-free, a binding mechanism is required to attach an instantiation of a type to a particular token. In the network, on the other hand, tokens are distinguished from one another by virtue of producing small but potentially discriminable differences in the state space. $John_{23}$, $John_{43}$, and $John_{192}$ (using subscripts to indicate different occurrences of the same lexical item) will be physically different vectors. Their identity as tokens of the same type is captured by the fact that they are all located in a region which may be designated as the *John* space, and which contains no other vectors. Thus, one can speak of this bounded region as corresponding to the lexical type, *John*.

The differences in context, however, create differences in the state. Furthermore, these differences are systematic. The clustering tree in Figure 3 was carried out over the mean vector for each word, averaged across contexts. If the actual hidden unit activation patterns are used, the tree is of course quite large since there are hundreds of tokens of each word. Inspection of the tree reveals two important facts. First, all tokens of a type are more similar to one another than to any other type, so the arborization of tokens of *boy* and *dog* do not mix (although, as was pointed out, such overlap is not impossible and may in some circumstances be desirable). Second, there is a substructure to the spatial distribution of tokens which is true of multiple types. Tokens of *boy* used as subject occur more closely to one another than to the tokens of *boy* as object. This is also true of the tokens of *girl*. Moreover, the spatial dimension along which subject-tokens vary from object-tokens is the same for all nouns. Subject-tokens of all nouns are positioned in the same region of this dimension, and object-tokens are positioned in a different region. This means that rather than proliferating an undesirable number of representations, this tokenization of types actually encodes grammatically relevant information. Note that the tokenization process does not involve creation of new syntactic or semantic atoms. It is, instead, a systematic process. The state space dimensions along which token variation occurs may be interpreted meaningfully. The token's location in state space is thus at least functionally compositional (in the sense described by van Gelder, 1990).

5.3 Polysemy and accommodation

Polysemy refers to the case where a word has multiple senses. Accommodation is used to describe the phenomenon in which word meanings are contextually altered (Langacker, 1987). The network approach to language processing provides an account for both phenomena, and shows how they may be related.

Although there are clear instances where the same phonological form has entirely different meanings (*bank*, for instance), in many cases polysemy is a matter of degree. There may be senses which are different, although metaphorically related, as in (5):

5. (a) Arturo Barrios runs very fast!

(b) This clock runs slow.

(c) My dad runs the grocery store down the block.

In other cases, the differences are far more subtle, though just as real:

6. (a) Frank Shorter runs the marathon faster than I ever will.

(b) The rabbit runs across the road.

(c) The young toddler runs to her mother.

In (6), the construal of *runs* is slightly different, depending on who is doing the running. But just as in (5), the way in which the verb is interpreted depends on context. As Langacker (1987) has described the process:

> It must be emphasized that syntagmatic combination involves more than the simple addition of components. A composite structure is an integrated system formed by coordinating its components in a specific, often elaborate manner. In fact, it often has properties that go beyond what one might expect from its components alone..... [O]ne component may need to be adjusted in certain details when integrated to form a composite structure; I refer to this as **accommodation**. For example, the meaning of run as applied to humans must be adjusted in certain respects when extended to four legged animals such as horses, dogs, and cats... in a technical sense, this extension creates a new **semantic variant** of the lexical item. (pp. 76-77).

In Figure 8 we see that the network's representations of words in context demonstrates just this sort of accommodation. Trajectories are shown for various sentences, all of which contain the main verb *burn*. The representation of the verb varies, depending on the subject noun. The simulations shown here do not exploit the variants of the verb, but it is clear that this is a basic property of such networks.

—Insert Figure 8 about here—

5.4 “Leaky recursion” and processing complex sentences

The sensitivity to context which is illustrated in Figure 8 also occurs across levels of organization. The network is able to represent constituent structure (in the form of embedded sentences), but it is also true that the representation of embedded elements may be affected by words at other syntactic levels.

This means that the network does not implement a stack or pushdown machine of the classical sort, and would seem not to implement true recursion, in which information at each level of processing is encapsulated and unaffected by information at other levels. Is this good or bad?

If one is designed a programming language, this sort of “leaky” recursion is highly undesirable. It is important that the value of variables local to one call of a procedure not be affected by their value at other levels. True recursion provides this sort of encapsulation of information. I would suggest that the appearance of a similar sort of recursion in natural language is deceptive, however, and that while natural language may require one aspect of what recursion provides (constituent structure and self-embedding) it may not require the sort of informational firewalls between levels of organization.

Indeed, embedded material typically has an elaborative function. Relative clauses, for example, provide information about the head of a noun phrase (which is at a higher level of organization). Adverbial clauses perform a similar function for main clause verbs. In general, then, subordination involves a conceptual dependence between clauses. Thus, it may be important that a language processing mechanism facilitate rather than impede interactions across levels of information.

There are specific consequences for processing which may be observed in a system of this sort, which only loosely approximates recursion. First, the finite bound on precision means that right-branching sentences (such as 7a) will be processed better than center-embedded sentences (such as 7b):

7. (a) The woman saw the boy that heard the man that left.

   (b) The man the boy the woman saw heard left.

It has been known for many years that sentences of the first sort are processed in humans more easily and accurately than sentences of the second kind, and a number of reasons have been suggested (e.g., Miller & Isard, 1964). In the case of the network, such an asymmetry arises because right-branching structures do not require that information be carried forward over embedded material, whereas in center-embedded sentences information from the matrix sentence must be saved over intervening embedded clauses.

But it is also true that not all center-embedded sentences are equally difficult to comprehend. Intelligibility may be improved in the presence of semantic constraints. Compare the following, in (8):

8. (a) The man the woman the boy saw heard left.

   (b) The claim the horse he entered in the race at the last minute was a ringer was absolutely false.

In (8b) the three subject nouns create strong—and different—expectations about possible verbs and objects. This semantic information might be expected to help the hearer more quickly resolve the possible subject/verb/object associations and assist processing (Bever, 1970; King & Just, 1991). The verbs in (8a), on the other hand, provide no such help. All three nouns might plausible be the subject of all three verbs.

In a series of simulations, Weckerly & Elman (1992) demonstrated that a simple recurrent network exhibited similar performance characteristics. It was better able to process right-branching structures, compared to center-embedded sentences. And center-embedded sentences which contained strong semantic constraints were processed better compared to center-embedded sentences without such constraints. Essentially, the presence of constraints meant that the internal state vectors generated during processing were more distinct (further apart in state space) and therefore preserved information better than the vectors in sentences in which nouns were more similar.

5.5 The immediate availability of lexically-specific information

One question which has generated considerable controversy concerns the time-course of processing, and when certain information may be available and used in the process of sentence processing. One proposal is that there is a first-pass parse during which only category-general syntactic information is available (Frazier & Rayner, 1982). The other major position is that considerably more information, including lexically-specific constraints on argument structure, is available and used in processing (Taraban & McClelland, 1990). Trueswell, Tanenhaus, and Kello (in press) present empirical evidence from a variety of experimental paradigms which strongly suggests that listeners are able to use subcategorization information in resolving the syntactic structure of a noun phrase which would otherwise be ambiguous. For example, in (9), the verb *forgot*permits both a noun phrase complement and also a sentential complement; at the point in time when *the solution* has been read, either (9a) or (9b) is possible.

9. (a) The student forgot the solution was in the back of the book.

 (b) The student forgot the solution.

In (10), on the other hand, *hope* is strongly biased toward taking a sentential complement.

10. (a) The student hoped the solution was in the back of the book.

(b) *The student hoped the solution.

Trueswell and his colleagues found that subjects appeared not only to be sensitive to the preferred complement for these verbs, but that behavior was significantly correlated with the statistical patterns of usage (determined through corpus analysis). That is, insofar as the actual usage of a verb might be more or less biased in a particular direction, subjects' expectations were more or less consistent with that usage. This is exactly the pattern of behavior which would be expected given the model of processing which has been described here, and we are currently attempting to model these data.

6 Conclusions

Over recent years, there has been considerable work in attempting to understand various aspects of speech and language in terms of dynamical systems. Some of the most elegant and well-developed work has focused on motor control, particularly within the domain of speech (e.g., Fowler, 1980; Kelso, Saltzman, & Tuller, 1986). Some of this work makes explicit reference to consequences for theories of phonology (e.g., Browman & Goldstein, 1985; Pierrehumbert & Pierrehumbert, 1990).

More recently, attention has been turned to systems which might operate at so-called higher-levels of language processing. One of the principal challenges has been whether or not these dynamical systems can deal in a satisfactory way with the apparently recursive nature of grammatical structure.

I have attempted to show in this chapter that indeed, networks which possess dynamical characteristics have a number of properties which capture important aspects of language, including their embedded nature. The framework appears to differ from traditional view of language processors in the way in which it represents lexical and grammatical information. Nonetheless, these networks exhibit behaviors which are highly relevant for language. They are able to induce lexical category structure from statistical regularities in usage; and they are able to represent constituent structure to a certain degree. They are not perfect, but their imperfections strongly resemble those observed in human language users.

Let me close, however, with an obvious caveat. None of the work described here qualifies as a full model of language use. The range of phenomena illustrated is suggestive, but limited. As any linguist will note, there are many, many questions which remain unanswered. The models are also disembodied in a way which makes it difficult to capture any natural semantic relationship with the world. These networks are essentially exclusively language processors and their language use is unconnected with an ecologically plausible activity. Finally, and related to prior point, the view of language use in these networks is deficient in that it is solely reactive. These networks are input/output devices. Given an input, they produce the output which is appropriate for that training regime. The networks are thus tightly coupled with the world in a manner which leaves little room for endogenously generated activity. There is no possibility here for either spontaneous speech or for reflective internal language. Put most bluntly, these are networks that do not think!

These same criticisms may be levelled, of course, at many other current and more traditional models of language, so they should not be taken as inherent deficiencies of the approach. Indeed, I suspect that the view of linguistic behavior as deriving from a dynamical system probably allows for greater opportunities for remedying these shortcomings. One exciting approach involves embedding such networks in environments in which their activity is subject to evolutionary pressure, and viewing them as examples of artificial life (e.g., Nolfi, Elman, &

Parisi, in press). But in any event, it is obvious that much remains to be done.

Guidelines for Further Reading

A good collection of recent work in connectionist models of language may be found in N. Sharkey (Ed.), *Connectionist Natural Language Processing: Readings from Connection Science.* Oxford: Intellect. The initial description of simple recurrent networks appears in Elman, J.L. (1990). Finding structure in time. *Cognitive Science,* 14, 179-211. Additional studies with SRNs are reported in Cleeremans, A., Servan-Schreiber, D., & McClelland, J.L. (1989). Finite state automata and simple recurrent networks. *Neural Computation,* 1, 372-381; and in Elman, J.L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7, 195-225. A discussion of recurrent networks as dynamical systems is found in Pollack, J.B. (1990). The induction of dynamical recognizers. *Machine Learning,* 7, 227-252.

References

Agre, P.E., & Chapman, D. (1987). Pengi: An implementation of a theory of activity. In *Proceedings of the AAAI-87.* Los Altos, CA: Morgan Kaufmann.

Altmann, G.T.M. & Steedman, M.J. (1988). Interaction with context during human sentence processing. *Cognition*, 30, 191-238.

Bever, T. (1970a). The cognitive basis for linguistic structure. In J.R. Hayes (Ed.), *Cognition and the development of language.* New York: Wiley.

Blaubergs, M.S. & Braine, M.D.S. (1974). Short-term memory limitations on decoding self-embedded sentences. *Journal of Experimental Psychology,* 102, No.4, 745-748.

Brooks, R.A. (1989). A robot that walks: Emergent behaviors from a carefully evolved network. *Neural Computation*, 1, 253-262.

Browman, C.P., & Goldstein, L. (1985). Dynamic modeling of phonetic structure. In V. Fromken (Ed.), *Phonetic linguistics.* New York: Academic Press.

Chomsky, N. (1975). *Reflections on Language.* New York: Pantheon.

Churchland, P.S., & Sejnowski, T.J. (1992). *The computational brain.* Cambridge, MA: MIT Press.

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

Elman, J.L., (1991a). Representation and structure in connectionist models. In Gerald Altmann (Ed.), *Computational and psycholinguistic approaches to speech processing.* New York: Academic Press.

Elman, J.L. (1991b). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7, 195-225.

Ferreira, F. & Henderson, J.M. (1990). The use of verb information in syntactic parsing: A comparison of evidence from eye movements and word-by-word self-paced reading. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 16, 555-568.

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

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

Forster, K. (1976). Accessing the mental lexicon. In R.J. Wales & E. Walker (Eds.), *New approaches to language mechanisms.* Amsterdam: North-Holland.

Fowler, C. (1977). *Timing and control in speech production.* Bloomington, IN: Indiana University Linguistics Club.

Fowler, C. (1980). Coarticulation and theories of extrinsic timing control. *Journal of Phonetics*, 8, 113-133.

Frazier, L. (1987). Sentence processing: A tutorial review. In M. Coltheart (Ed.), *Attention and Performance XII: The psychology of reading.* Hillsdale, NJ: Erlbaum.

Frazier, L., & Rayner, K. (1982). Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology,* 14, 178-210.

Harris, C. & Elman, J.L. (1989). Representing variable information with simple recurrent networks. In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

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

Jordan, M.I., & Rosenbaum, D.A. (1988). Action. Technical Report 88-26.

Department of Computer Science, University of Massachusetts at Amherst.

Kelso, J.A.S., Saltzman, E., & Tuller, B. (1986). The dynamical theory of speech production: Data and theory. *Journal of Phonetics*, 14, 29-60.

King, J & Just, M.A. (1991). Individual differences in syntactic processing: the role of working memory. *Journal of Memory and Language*, 30, 580-602.

Langacker, R.W. (1987). *Foundations of cognitive grammar: Theoretical perspectives.* Volume 1. Stanford: Stanford University Press.

MacNeilage, P.F. (1970). Motor control of serial ordering of speech. *Psychological Review*, 77, 182-196.

Marslen-Wilson, W.D. (1980). Speech understanding as a psychological process. In J.C. Simon (Ed.), *Spoken language understanding and generation.* Dordrecht: Reidel.g

McClelland, J.L., & Elman, J.L. (1986). The TRACE model of speech perception. *Cognitive Psychology,* 18, 1-86.

McClelland, J.L., & Rumelhart, D.E. (1981). An interactive activation model of contexts effects in letter perception: Part 1. An account of basic findings. *Psychological Review*, 88, 365-407.

Miller, G.A., & Chomsky, N. (1963). Finitary models of language users. In R.D. Luce, R.R. Bush, & E. Galanter (Eds.), *Handbook of mathematical psychology* (Vol. II). New York: Wiley.

Miller, G. & Isard, S. (1964). Free recall of self-embedded English sentences. *Information and Control*, 7, 292-303.

Morton, J. (1979). Word recognition. In J. Morton & J.C. Marshall (Eds.), *Psycholinguistics 2: Structures and processes.* Cambridge, MA: MIT Press.

Mozer, M.C. (1989). A focused back-propagation algorithm for temporal pattern recognition. *Complex Systems*, 3, 49-81.

Nolfi, S., Elman, J.L., & Parisi, D. (in press). Learning and evolution in neural networks. *Adaptive Behavior.*

Pearlmutter, B.A. (1989). Learning state space trajectories in recurrent neural networks. *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C., II-365.

Pierrehumbert, J.B., & Pierrehumbert, R.T. (1990). On attributing grammars to dynamical systems. *Journal of Phonetics*, 18, 465-477.

Pollack, J.B. (1990). The induction of dynamical recognizers. *Machine Learning*, 7, 227-252.

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.

Selfridge, O.G. (1958). Pandemonium: A paradigm for learning. *Mechanisation of thought processes: Proceedings of a symposium held at the National Physical Laboratory, November 1958*. London: HMSO.

Siegelmann, H.T., & Sontag, E.D. (1992). Neural networks with real weights: Analog computational complexity. Report SYCON-92-05. Rutgers Center for Systems and Control, Rutgers University.

Simon, H. (1980). Physical symbol systems. *Cognitive Science*, 4, 135-183.

St. John, M. F. (1992). The story gestalt: A model of knowledge-intensive processes in text comprehension. *Cognitive Science*, 16, 271-306.

St. John, M., & McClelland, J.L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence*, 46, 217-457.

Taraban, R., & McClelland, J.L. (1988). Constituent attachment and thematic role expectations. *Journal of Memory and Language*, 27, 597-632.

Trueswell, J.C., Tanenhaus, M.K., & Kello, C. (in press). Verb-specific constraints in sentence processing: Separating effects of lexical preference from garden-paths. *Journal of Experimental Psychology: Learning, Memory and Cognition*.

Van Gelder, T. (1990). A connectionist variation on a classical theme. *Cognitive Sciece*, 14, 355-384.

Weckerly, J., & Elman, J.L. (1992). A PDP approach to processing center-embedded sentences. *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum.

Wiles, J., & Bloesch, A. (1992). Operators and curried functions: Training and analysis of simple recurrent networks. In J.E. Moody, S.J. Hanson, & R.P. Lippman (Eds.), *Advances in Neural Information Processing Systems 4.* San Mateo, CA: Morgan Kaufmann.

Figure Legends

Figure 1. A feed-forward network which represents time through space. Circles represent nodes; arrows between layers indicate full connectivity between nodes in adjacent layers. The network is feed-forward because activations at each level depend only on the input received from below. At the conclusion of processing an input, all activations are thus lost. A sequence of inputs can be represented in such an architecture by associating the first node (on the left) with the first element in the sequence; the second node with the second element; and so on.

Figure 2. A simple recurrent network (SRN). Solid lines indicate full connectivity between layers, with weights which are trainable. The dotted line indicates a fixed one-to-one connection between hidden and context layers. The context units are used to save the activations of the hidden units on any time step. Then on the next time step, the hidden units are activated not only by new input but by the information in the context units—which is just the hidden units' own activations on the prior time step. An input sequence is processed by presenting each element in the sequence one at a time, allowing the network to be activated at each step in time, and then proceeding to the next element. Note that although hidden unit activations may depend on prior inputs, by virtue of prior inputs' effects on the recycled hidden unit/context unit activations, the hidden units do not record the input sequence in any veridical manner. Instead, the task of the network is to learn to encode temporal events in some more abstract manner which allows the network to perform the task at hand.

Figure 3. Hierarchical clustering diagram of hidden units activations in the

simulation with simple sentences. After training, sentences are passed through the network, and the hidden unit activation pattern for each word is recorded. The clustering diagram indicates the similarity structure among these patterns. This structure, which reflects the grammatical factors that influence word position, is inferred by the network; the patterns which represent the actual inputs are orthogonal and carry none of this information.

Figure 4. The predictions made by the network in the simulation with complex sentences, as the network processes the sentence “boys who Mary chases feed cats.” Each panel displays the activations of output units after successive words; outputs are summed across groups for purposes of displaying the data. “V” and “N” refer to verbs and nouns; “sg” and “pl” refer to singular and plural; “prop” refers to proper nouns; and “t”, “i”, and “t/i” refer to transitive verbs, intransitive verbs, and optionally transitive verbs.

Figure 5. Trajectories through hidden unit state space as the network processes the sentences “boy hears boy” and “boys hear boy”. The number (singular vs. plural) of the subject is indicated by the position in state space along the second principal component.

Figure 6. Trajectories through hidden unit state space as the network processes the sentences “boy chases boy”, “boy sees boy”, and “boy walks.” Transitivity of the verb is encoded by is position along an axis which cuts across the first and third principal components.

Figure 7. Trajectories through hidden units state space (principal components 1 and 11) as the network processes the sentences “boy chases boy”, “boy chases boy who chases boy”, “boy who chases boy chases boy”, and “boy chases boy who chases boy who chases boy” (to assist in reading the plots, the final word of each sentence is terminated with a “]S”).

Figure 8. Trajectories through hidden units state space (principal components 1

and 2) as the network processes the sentences “{john, mary, lion, tiger, boy, girl} burns house”, as well as “{museum, house} burns” (the final word of each sentences is terminated with “]S”). The internal representations of the word “burns” varies slightly as a function of the verb’s subject.

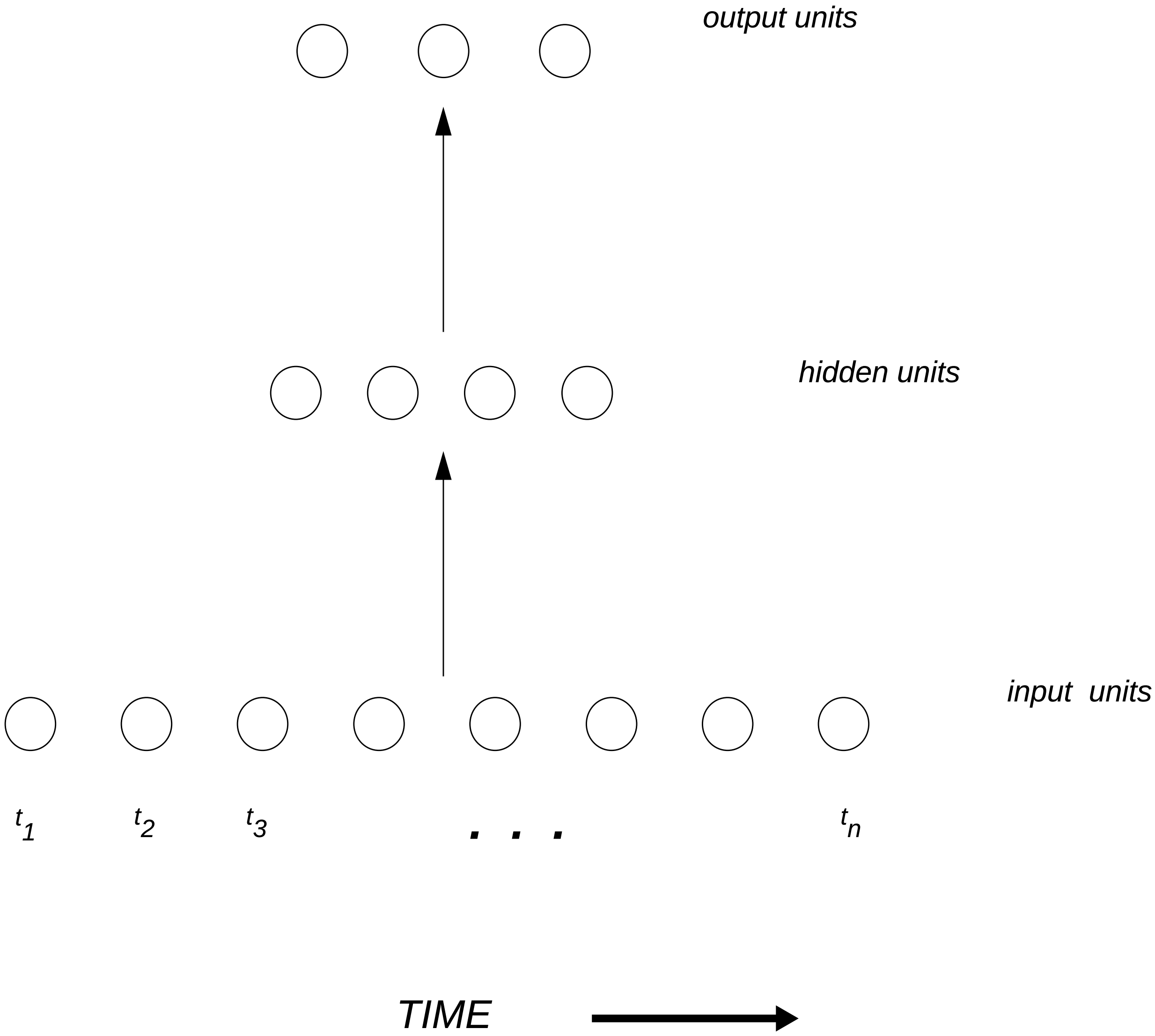

Figure 1

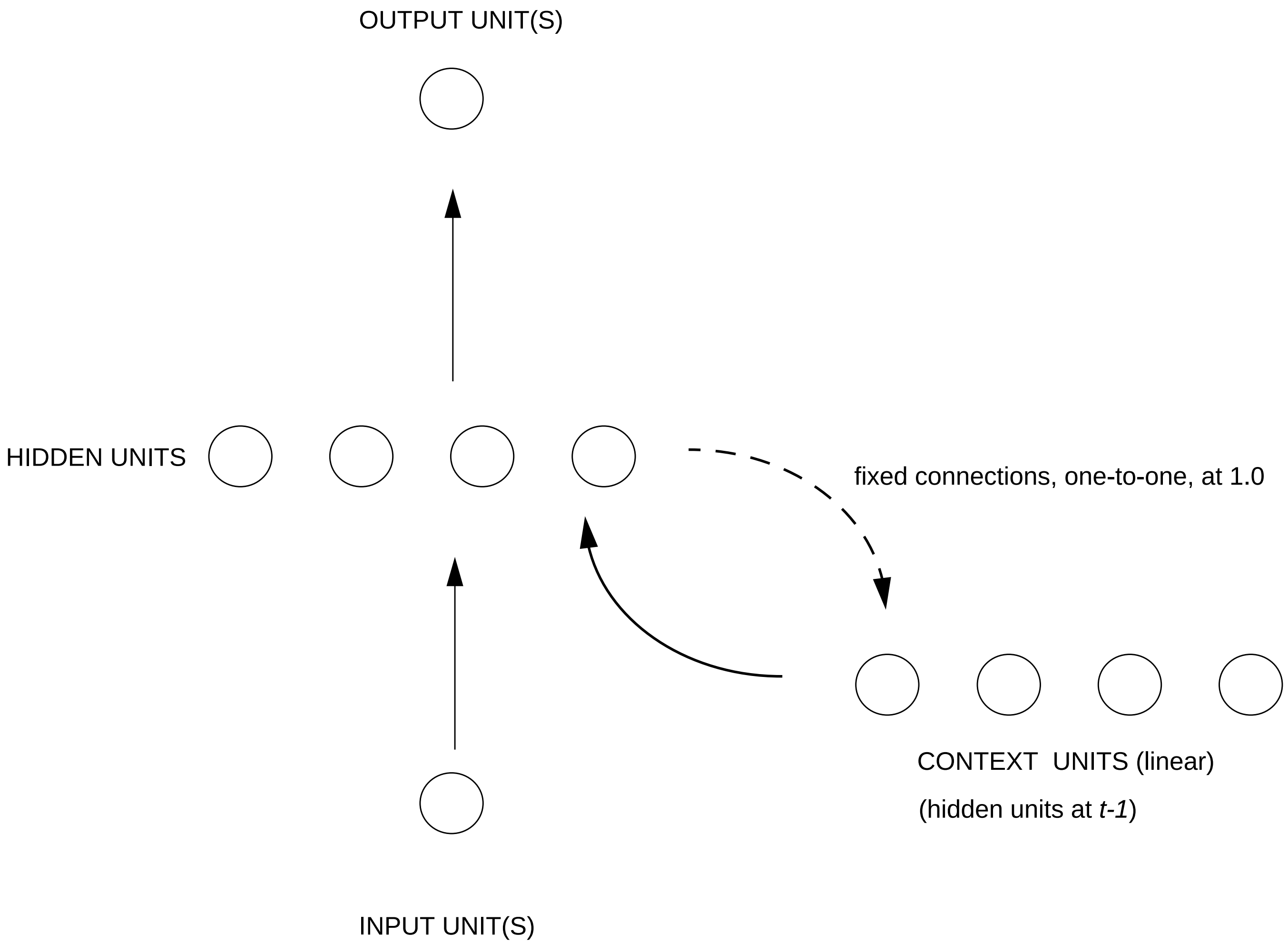
OUTPUT UNIT(S)
HIDDEN UNITS
fixed connections, one-to-one, at 1.0
CONTEXT UNITS (linear)
(hidden units at *t-1*)
INPUT UNIT(S)

Figure 2

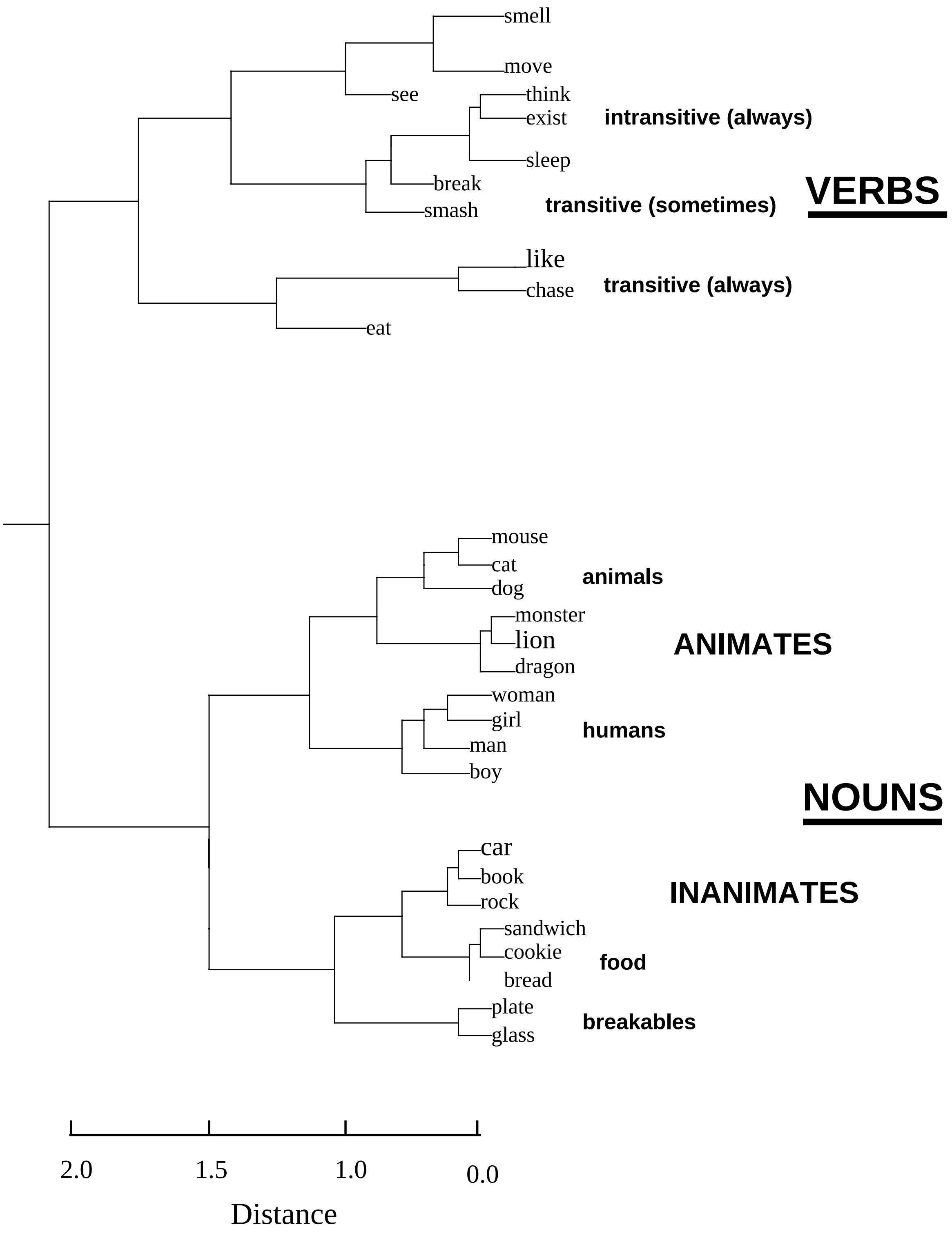
smell
move
see
think
exist
intransitive (always)
sleep
break
VERBS
smash
transitive (sometimes)
like
chase
transitive (always)
eat
mouse
cat
animals
dog
monster
lion
ANIMATES
dragon
woman
girl
humans
man
boy
NOUNS
car
book
INANIMATES
rock
sandwich
cookie
food
bread
plate
breakables
glass
2.0
1.5
1.0
0.0
Distance

Figure 3

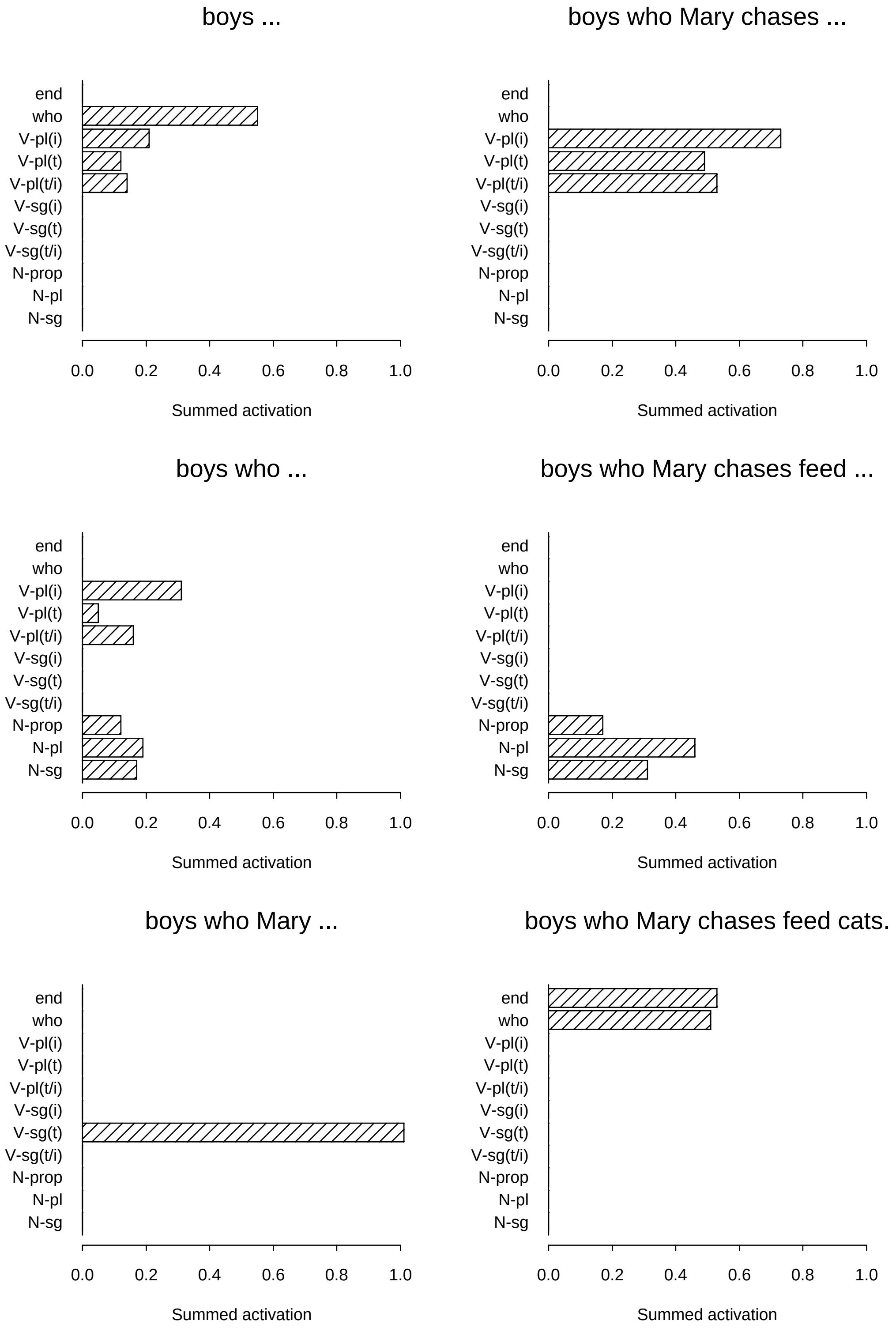
boys ...
boys who Mary chases ...
boys who ...
boys who Mary chases feed ...
boys who Mary ...
boys who Mary chases feed cats.
end
who
V-pl(i)
V-pl(t)
V-pl(t/i)
V-sg(i)
V-sg(t)
V-sg(t/i)
N-prop
N-pl
N-sg
0.0
0.2
0.4
0.6
0.8
1.0
Summed activation

Figure 4

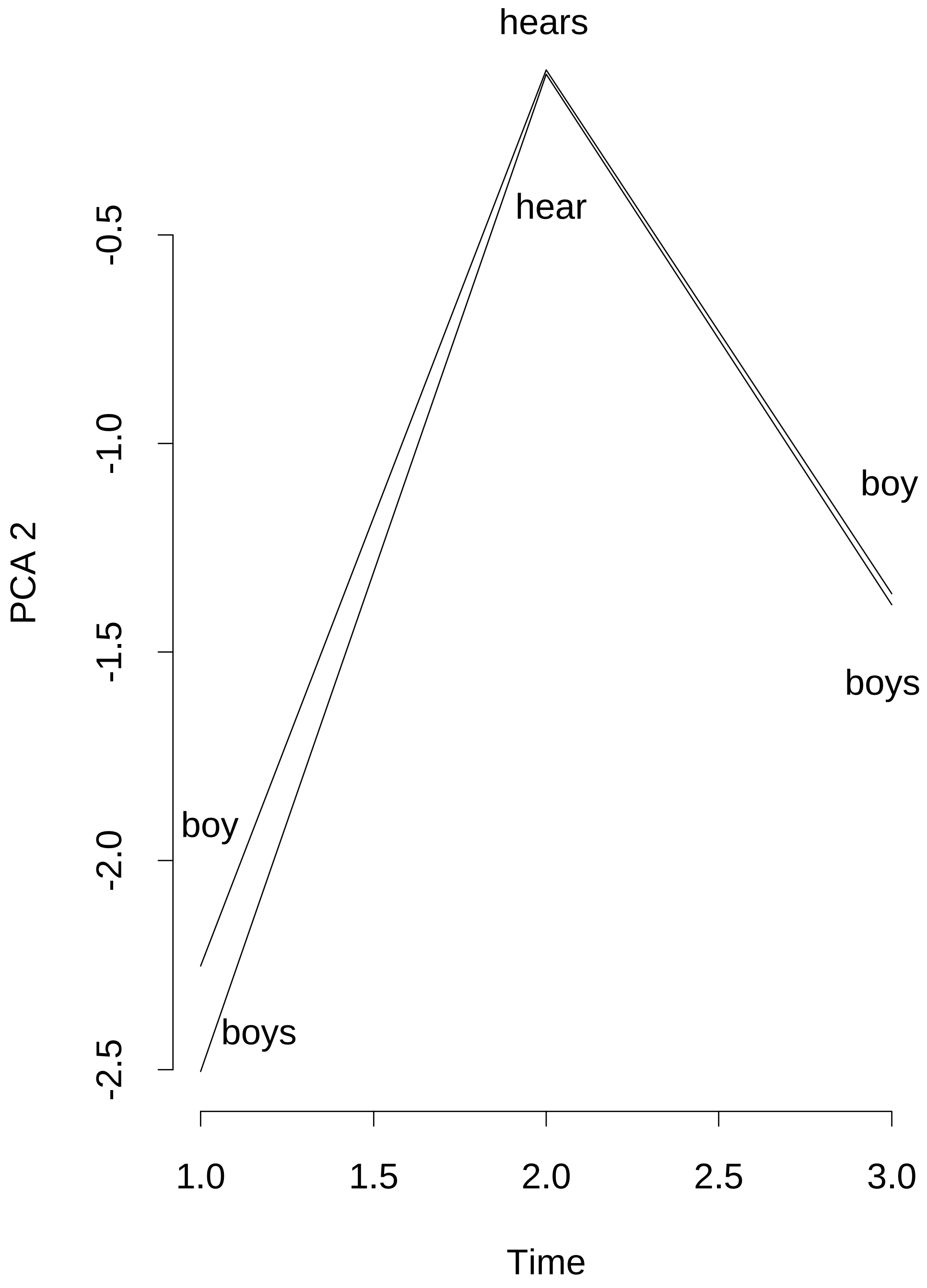
hears
hear
boy
boys
boy
boys
PCA 2
-0.5
-1.0
-1.5
-2.0
-2.5
1.0
1.5
2.0
2.5
3.0
Time

Figure 5

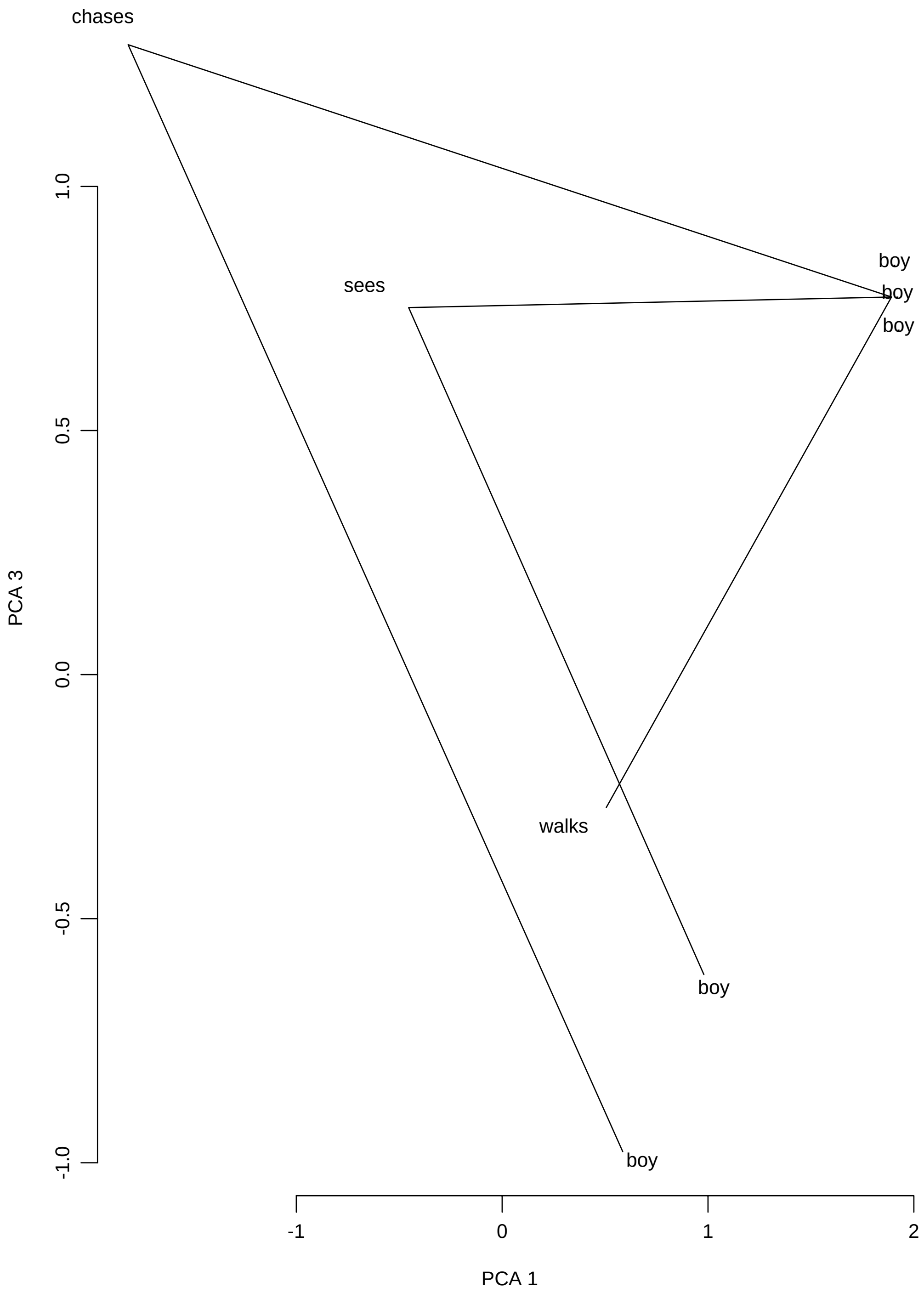
chases
sees
boy
boy
boy
walks
boy
boy
1.0
0.5
0.0
-0.5
-1.0
PCA 3
-1
0
1
2
PCA 1

Figure 6

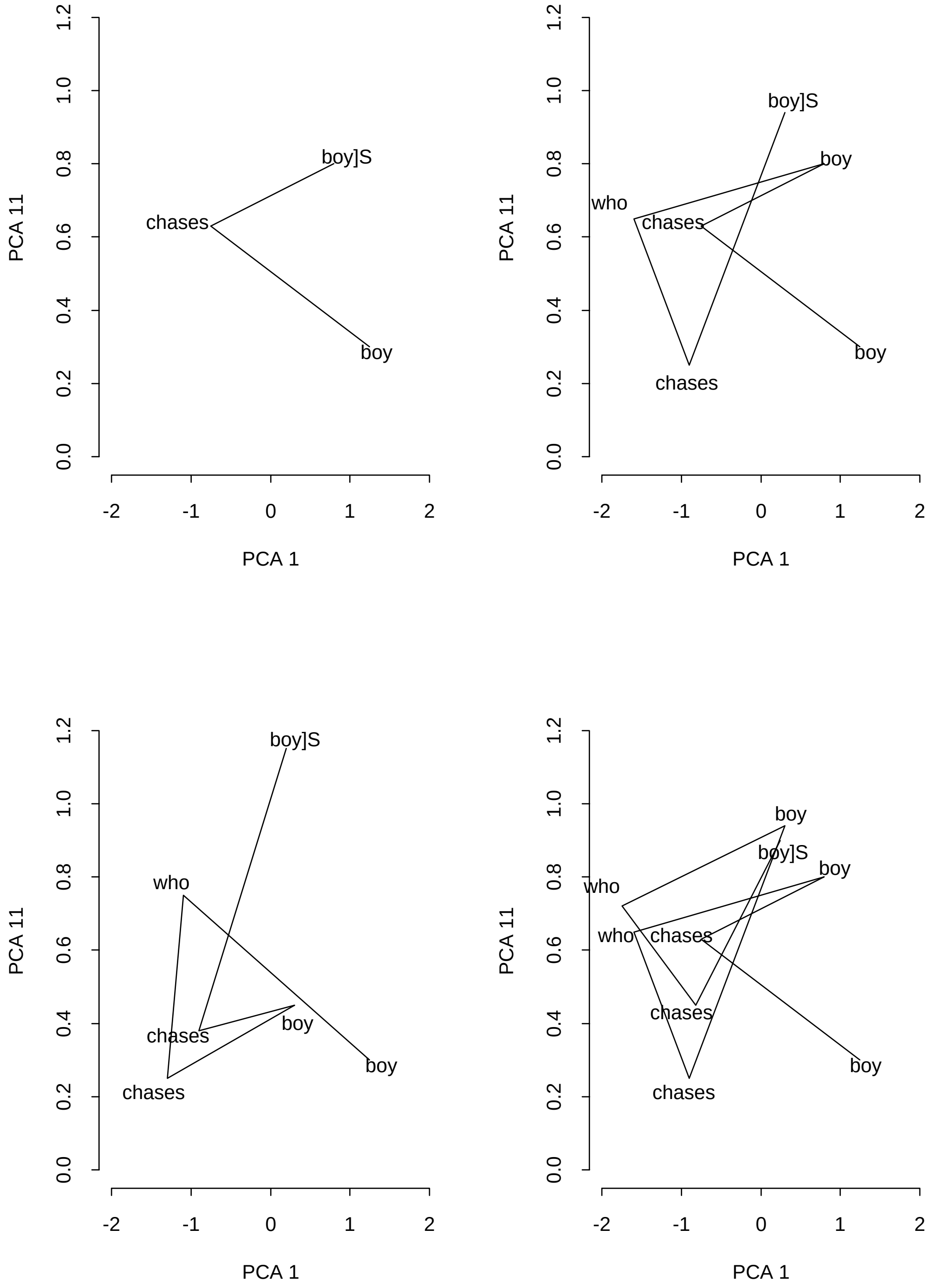
PCA 11
PCA 1
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-2
-1
0
1
2
boy]S
chases
boy
boy]S
boy
who
chases
chases
boy
boy]S
who
chases
boy
chases
boy
boy
boy]S
boy
who
who
chases
chases
chases
boy

Figure 7

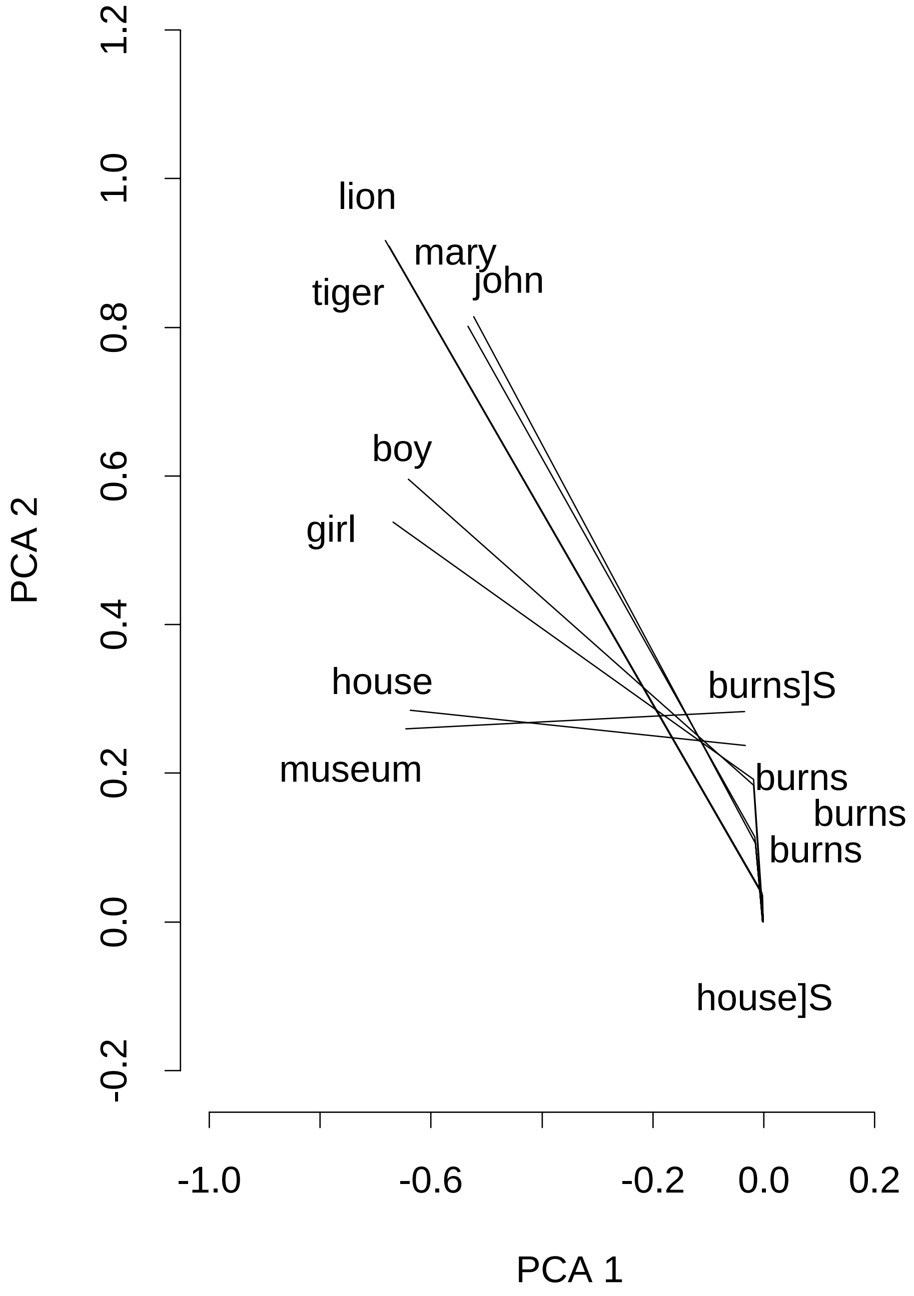
lion
mary
john
tiger
boy
girl
house
museum
burns]S
burns
burns
burns
house]S
PCA 2
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-1.0
-0.6
-0.2
0.0
0.2
PCA 1

Figure 8