Sentence Processing and Linguistic Structure

Whitney Tabor

tabor@uconnvm.uconn.edu

University of Connecticut

In Press in Kremer, S. and Kolen, J. (Eds.),
Field Guide to Dynamical Recurrent Networks. IEEE.

February 14, 2001

1 Introduction

Dynamical systems theory provides effective formal models of structure in natural languages. We describe a recurrent neural network called the Bramble Network (BRN) and a related analytical tool, the Dynamical Automaton (DA), which process words in sequence. The BRN makes one discrete jump across its state space each time a word is processed, and then settles continuously to a stable state. Processing time is modeled as convergence time. Two well-known phenomena in natural language processing are modeled: (i) the inverse correlation between word frequency and response time and (ii) the correlation between parsing difficulty and level of center embedding. The model shows how constructs of dynamical systems theory provide a link between neural network models which are good at learning and show human-like flexibility and abstract linguistic representations which are the current best model of natural language syntactic structure and interpretation.

What role does dynamical systems theory have to play in turn-of-the-millenium Cognitive Science? To answer this question, it is helpful to identify two major contrasting viewpoints that have come to prominence in the past century. One viewpoint, identified with the work of J. B. Watson, B.F. Skinner and other *behaviorists*, holds that all cognition can be derived by studying experimentally-detected relationships between the inputs and outputs of an organism and that speculation about unobservable "mental" processes is fruitless. The other viewpoint, associated especially with N. Chomsky, A. Newell, M. Minsky, J. Fodor and other *cognitivists* urges the researcher to start with a theory of mental entities (e.g. memory, imagery, grammar, etc.), and then test the predictions of that theory empirically. The latter view has flourished recently. Since the 1950s there has been a vigorous industry of investigating mental entities in psychology and linguistics. Although behaviorism has fallen into disfavor since the debates, it has a cousin in the form of *connectionist* or *neural network* models of cognition. Connectionist learning models have the advantage that they can do a lot with the simple "stimulus-response" rubric of behaviorism and are grounded in an appropriately profound mathematics (contrast both behaviorism and traditional artificial intelligence research). On the other hand, they inherit one of behaviorism's weaknesses: opacity. It is often easy to get a network to behave in an appealing way, at least on simple tasks; it is much harder to understand in a principled sense why it behaves that way and to get it to

learn more complex tasks.

Dynamical systems theory can help us understand the organizing principles of connectionist networks by clarifying the relationship between two perspectives on complex systems: the *topological* and the *metric*. The topological perspective groups many states of a system together into basins associated with similar long-term behavior and thus sheds light on the system's large-scale, organizing properties. The metric perspective emphasizes the details of short-term behavior, thus relating large-scale structures to a timescale that is suitable for modeling the daily lives of people and animals. Virtually all connectionist networks are dynamical systems in the classical sense because their feedback mechanisms (the learning feedback loop in some cases, the relaxation feedback loop in others, or both) give rise to attractor structures. The metric perspective dominated early connectionist research. The topological perspective is now being more fully developed.

In fact, many of the fundamental constructs of dynamical systems theory (basins, fixed points, stability) have been put forth as useful models of cognition (Chapter 2 of this volume; (van Gelder, 1998)). There are also a variety of complex structures that arise in high-dimensional and/or chaotic dynamical systems. The possibility that some of the latter structures will provide insight into the useful but often vaguely specified mental entities identified by cognitive psychologists and linguists makes the intersection of dynamics and cognition particularly interesting to explore at present.

1.1 The role of recurrent connectionist networks

Recurrent connectionist networks and variants on them have been used in a variety of cognitive domains: language (Elman, 1990; Elman, 1991; Plaut et al., 1996; Rueckl, 1995; Tabor et al., 1997; Rodriguez et al., 1999; Tabor and Tanenhaus, 1999); music (Large and Kolen, 1999; Page, 1999); robotics (Schöner et al., 1995; Tani, 1998; Tani and Nolfi, 1999; Bergener et al., 1999). The special property of *dynamical recurrent networks*, in the strict sense of networks which settle in continuous time and connected space, is that they make the dynamical structures more explicit than feedforward networks or discrete-transition recurrent networks.

Elman (Elman, 1990; Elman, 1991) designed the Simple Recurrent Network (SRN) for sequence prediction. The SRN is essentially a discrete 3-

layer feedforward network[1] (input → hidden → output), but it has recurrent connections among the hidden units[2]. Elman used the network to study symbolic sequence prediction by adopting a localist representation of symbols on the input layer and training the network on the task of predicting the next symbol on the output layer, using a variant of Backpropagation Through Time (BPTT—see (Williams and Peng, 1990; Pearlmutter, 1995), Chapter **??** of this volume). The recurrent connections permit the SRN to encode arbitrary temporal dependencies, thus opening the door to learning infinite-state languages, like the context-free languages discussed in Section 2.2 below (see (Kremer, 1996)). Rodriguez, Wiles, and Elman (Rodriguez et al., 1999) analyzed a version of the SRN trained on the language $a^n b^n$ (n = 1, 2, 3, ...) and found that its computations took place on the transients of fixed points of the dynamical system associated with the recurrent hidden connections. In effect, the attractors provided the structure which organized the recursive computation (see Tabor, in press). It has proved more difficult to get the SRN to learn more complex languages—it is challenged by long-distance dependencies because of the diffusion of the error signal over successive layers in the BPTT regime. But Servan-Schreiber, Cleeremans, and McClelland (Servan-Schreiber et al., 1991), found that the network was much helped by the inclusion in the training corpus of subtle probabilistic biases that preserved state distinctions over the course of long dependencies. Rohde and Plaut (Rohde and Plaut, 1999) noted that semantic biases in natural language induce such subtle probabilistic biases in training corpora and they showed that the inclusion of these biases significantly improved SRN learning on a natural language task involving multiple center-embeddings. Thus SRNs can easily learn at least some classes of complex languages.

Plaut et al. (Plaut et al., 1996) studied a 3-layer attractor network for individual word naming. In their network, the input units encode an orthographic representation of a word. The inputs feed forward to a hidden layer which feeds forward to an output layer. The ouput units, which are supposed to produce a phonemic representation of the word represented on the input layer, are self-connected and also send feedback to the hidden units. The unit activations evolve continuously over time and are trained using conti-

[1]By a discrete network or dynamical system we mean a model in which instantaneous state changes are significantly noninfinitismal.

[2]The SRN is described in more detail in Section 2 below.

nous BPTT (Pearlmutter, 1989). Plaut et al. found that convergence time provided an accurate model of naming time in their model. In particular, their network exhibited the empirically established *frequency* $\times$ *regularity* interaction for individual-word naming: being low instead of high in frequency slows naming significantly for irregularly spelled words but does not make a difference for regularly spelled words (Seidenberg and McClelland, 1989). In our investigation of the Bramble Network (BRN) described below, we make the same analogy between network convergence time and human processing time. Plaut et al.'s model uses a slot-filler representation for sequences of symbols on the input layer. Natural language syntax, which seems best modeled by context-free and other infinite-state computational mechanisms, is not expressible in a slot-filler notation. Thus, while Plaut et al.'s model has the advantage of providing an explicit model of processing time, it lacks the SRN's capacity for modeling the temporal dependencies that arise in syntax. The BRN, described below, combines continuous settling with an SRN to obtain the advantages of both systems.

Tabor et al. (Tabor et al., 1997) and Tabor and Tanenhaus (Tabor and Tanenhaus, 1999) describe another extension of the SRN which makes the dynamics of word-by-word language processing explicit. Their model, called the Visitation Set Gravitation (VSG) model, combines an SRN with a dynamical system based on physical gravitation. The SRN's internal states are treated as inducing a mass density function, where more frequently visited regions are associated with greater concentrations of mass. A generalized form of the Law of Universal Gravitation models the movement of the system in the mass distribution. The system forms attractors corresponding to structures that arise in parsing and makes empirically supported predictions about the effects of word frequency and other sentence processing phenomena (Tabor and Tanenhaus, 1999). However, it is a complex hybrid model and thus requires some special tuning which can be avoided in a more uniform model. The BRN, described below, is a more uniform model which makes similar predictions to the VSG model and systematizes the relationship between dynamical structures and learning.

Large and Kolen (Large and Kolen, 1999) study a model of musical rhythm perception based on coupled oscillators. Their oscillators can be thought of as connectionist units with sinusoidal resting functions. Each oscillator has its period and phase set at particular values initially. In the presence of a rhythmic stimulus, the value of a coupling term is adjusted in

order to increase the alignment between the energy distribution of the stimulus and the energy distribution of the oscillator. Large and Kolen suggest that stability analyses, in the form of regime diagrams, provide the kind of insight into the structure of their system that well-formedness rules provide in more traditional metrical theories of rhythm. They identify the general case of interest in which the oscillators, like neurons, interact not just with the stimulus, but with each other, and then focus on the simple case in which such interoscillator-communication is not included. Their model succeeds in identifying fundamental periodicities in one complex musical passage. The most obvious analogy between music and natural language syntax is the one which treats melodic phrases as analogues of linguistic phrases, but the use of attractors to model components of a rhythm suggests an intriguing new way of thinking about phrasing in general: syntax might be a system of oscillators, where each word produced or encountered causes the system to calibrate an oscillator corresponding to the grammatical interpretation of the word.

Tani (Tani, 1998) studied a seeing robot that learned to navigate a cyclic course connecting five landmarks. At the core of the robot is a discrete recurrent network employing *context re-entry* (Jordan and Rumelhart, 1992). Input units encode the current scene and feed forward to hidden units. The hidden units, in turn, feed forward to output units which encode a prediction of the expected next scene. The hidden units also predict an output context which serves as input to the hidden units on the next timestep. The network alternates between two modes: an *open-loop mode*, in which it attempts to predict successive scenes while traveling in a room, and a *closed-loop mode*, in which it uses its outputs to update its inputs on the next timestep (and does not interact with the external world). Tani found that the system alternated somewhat spontaneously between chaotic and nonchaotic regimes. The nonchaotic regime often took the form of a five-period limit cycle whose states corresponded to the five landmarks on the robot's path. When a sixth landmark was added to the environment, the robot went into an especially long chaotic phase before developing a suitable period-six limit cycle. The results suggest that the chaos plays an important role in modification of system structure.

One important difference exists between the way a neural network solves a task and the way a symbolic computer solves it. A network fits its training cases with a connected manifold and generalizes by interpolation/extrapolation

on this manifold, while a symbolic computer defines categories that cover novel cases, but makes no assumption of representational continuity. In the above studies, the effect of taking a dynamical perspective has often been to enhance understanding of the in-between-the-datapoints structure of the network manifold, thus clarifying what is new about the network perspective.

1.2 The Role of Sentence Processing Studies

At present, only a few forays have been made into the domain of modeling complex natural language syntactic structure with self-organizing mechanisms like neural networks (Pollack, 1990; Elman, 1990; Elman, 1991; Elman, 1995; Port and van Gelder, 1995; Burgess and Lund, 1997; Landauer and Dumais, 1997; Tabor et al., 1997; Tabor and Tanenhaus, 1999; Rohde and Plaut, 1999). In several of these cases, a network is trained to make accurate predictions about a linguistic domain, the internal representations are analyzed, and the resulting cluster-structure bears a tantalizing resemblance to some of the abstractions that linguists have proposed as rudimentary constructs in language. For example, Elman (Elman, 1990) trained a network to predict successive words in small English-like corpus ("dog eat food . window break . girl see boy . girl break window . boy sleep . ..."). A hierarchical cluster analysis of the network's hidden units showed clusters corresponding to *Noun*, *Verb*, *Transitive Verb*, *Intransitive Verb*, *Animate*, *Inanimate*, etc. But in most such cases, the mapping to linguistic theory was rather vague, and there was a lack of theoretical completeness to the analysis. For example, What is the appropriate way to define clusters? Why do the clusters line up with linguistic-like categories?

The study of real-time language processing provides a more quantifiable angle on the nature of linguistic representation. This field was inspired by the global coherence that Generative Linguistic Theory brought to the study of natural language syntax in the 1960s. A variety of laboratory-based methods have been developed which allow researchers to make statistically robust claims about how people produce and comprehend language. The methods include measuring reaction times to linguistic stimuli—in particular, word-by-word reading times, tracking eye-movements in reading and visual scenes, measuring people's statistical biases in making choices between alternative syntactic structures when speaking or writing, and correlating various brain-imaging techniques with the presentation of linguistic stimuli. The goal of

all of these studies is to model human behavior as accurately as possible in order to gain insight into how the brain manages to interpret and produce language at the rapid rate that it does. In the work we describe below, we strive for a similar goal, focusing on the results of particular reading-time studies.

A number of connectionist networks have been successfully used to model sentence processing data—e.g., (Cottrell and Small, 1984; Selman and Hirst, 1985; McClelland and Kawamoto, 1986; Kempen and Vosse, 1989; Elman, 1990; Elman, 1991; Christiansen and Chater, 1994; Juliano and Tanenhaus, 1994; Christiansen and Chater, 1999; Vosse and Kempen, 1999). Many of these implicitly dynamical models have generally produced compelling quantitative fits to data but have been hard to understand and thus hard to make analytic predictions from.

1.3 Overview

Here, we show how the explicit introduction of dynamics and dynamical constructs into neural networks for symbol prediction provides explanation for two central sentence processing phenomena: Frequency sensitivity and Phrase-structure.

Frequency sensitivity refers to the way processing time measures are correlated with frequencies of abstract linguistic types. For example, the word "cinnamon" is an abstract entity which occurs some number of times in each sample of speech data that we may consider. It is well established that individual word reading times (both in isolation and in sentence context) are significantly correlated with the average rate of use of the words in natural speech (Inhoff and Rayner, 1986; Rayner and Duffy, 1986). Frequency effects are thus a benchmark phenomenon which a model of sentence processing should be able to predict.

Phrase structure and memory. One insight of modern linguistic theory has been that phrasal units (e.g. Noun Phrase, Verb Phrase, Adjective Phrase, Prepositional Phrase) are important organizing devices. Thus, symbolic methods of encoding natural language typically make use of context-free grammars (CFGs), which define languages as recursive concatenations of phrasal units. Neural network models that learn languages are taxed by the problem of learning context free languages (CFLs) because it is hard for them to detect long-distance temporal dependencies. Thus, simulation

studies of language processing, including the work on phenomena related to frequency of words which we describe below, have largely side-stepped the challenge of modeling phrasal organization (there are many interesting phenomena in language processing that can be studied using samples of language with only short temporal dependencies). In the larger picture, however, this bias creates an unfortunate lacuna because phrase representation is a powerful tool for encoding the large number of combinatorial possibilities that natural languages allow.

A distinct line of work has focused on the computational capacity of artificial neural networks, showing ways of implementing various infinite-state devices (like CFGs) in neural hardware (Siegelmann and Sontag, 1991; Kremer, 1996; Siegelmann, 1996). One species of this work (Barnsley, 1993; Moore, 1998; Tabor, 1998; Tabor, 2000) provides insight into what abstract phrasal structures look like in metric representation spaces by showing how they can arise via the interaction of attractors in a dynamical system. We descibe an application of this work to the modeling of sentence processing phenomena to demonstrate its empirical plausibility and to strengthen, for this critical case, the claim that dynamical systems theory identifies appropriate abstractions for handling complex cognitive phenomena.

2 Case studies: Dynamical networks for sentence processing

For the first study, we used the dynamical recurrent network shown in Figure 1. This network, called the *Bramble Network* (BRN), is based on the Simple Recurrent Network (SRN) architecture first investigated by Elman (Elman, 1990; Elman, 1991). We studied its performance on the same task that Elman investigated: next-word-prediction. In Elman's network, words are assigned localist encodings on an input layer. There are feedforward connections from input to hidden units and from hidden to output units, and there are also recurrent connections among the hidden units. These recurrent connections are implemented by treating the previous timestep of the hidden layer as an extra row of input units. These extra input units, called *Context Units* receive a copy of the previous hidden activations each time a new input is presented. Hidden units and output units are then updated according to

the discrete map,

$$a_i = \sigma_D(net_{a_i}) \tag{1}$$

$$net_{a_i} = \sum_j w_{ij} a_j \tag{2}$$

where a_i is the activation of unit i, $\sigma_D(x)$ is the logistic activation function, $1/(1+e^{-x})$, and w_{ij} is the weight from unit j to unit i.

The BRN has two parallel sets of recurrent connections among its hidden units. The first set, called the *discrete weights*, are equivalent to the weights from the context units to the hidden layer in Elman's network. The second set, called the *continuous weights*, are used for settling in connected space as specified in Equation 3.

$$\frac{dv_i}{dt} = net_i - v_i \tag{3}$$

Here, $net_i = b_i + \sum_j w_{ij}\sigma_C(v_j)$, $\sigma_C(x) = tanh(x)$, and $tanh(v_i) = 2 * a_i - 1$. Because it is designed for performing a one-in-n prediction task, the BRN we used here has normalized exponential (or softmax) output units ($\sigma_i = \frac{e^{net_{a_i}}}{\sum_{Outputs} e^{net_{a_j}}}$). In the BRN, the input and previous-time-step hidden units are updated first. Then the input-to-hidden weights and the discrete hidden-to-hidden weights are used to compute an initial new state of the hidden units. Continuous settling is carried out (depending on what was being modeled, it was carried out to convergence, or for a prespecified amount of time) via the continuous weights among the hidden units. Finally, the hidden-to-output weights map the final state of the hidden units to the output.

The network is trained using backpropagation (Rumelhart et al., 1986; Rumelhart et al., 1995) assuming a multinomial cost function ($E_D = \log \Pi_{Outputs} y_j^{t_i}$). The discrete weights are trained using discrete BPTT, but as in Elman's studies, the gradient was truncated after a single timestep (called BPTT(1) by (Williams and Peng, 1990); see also Chapter **??** of this volume).

While the discrete training accomplishes *accuracy maximization*, the continuous weights are updated according to a principle of *stability maximization*. That is, for continuous weights, we define the error on unit i as

$$E_{Ci} = \left(\frac{dv_i}{dt}\right)^2 \tag{4}$$

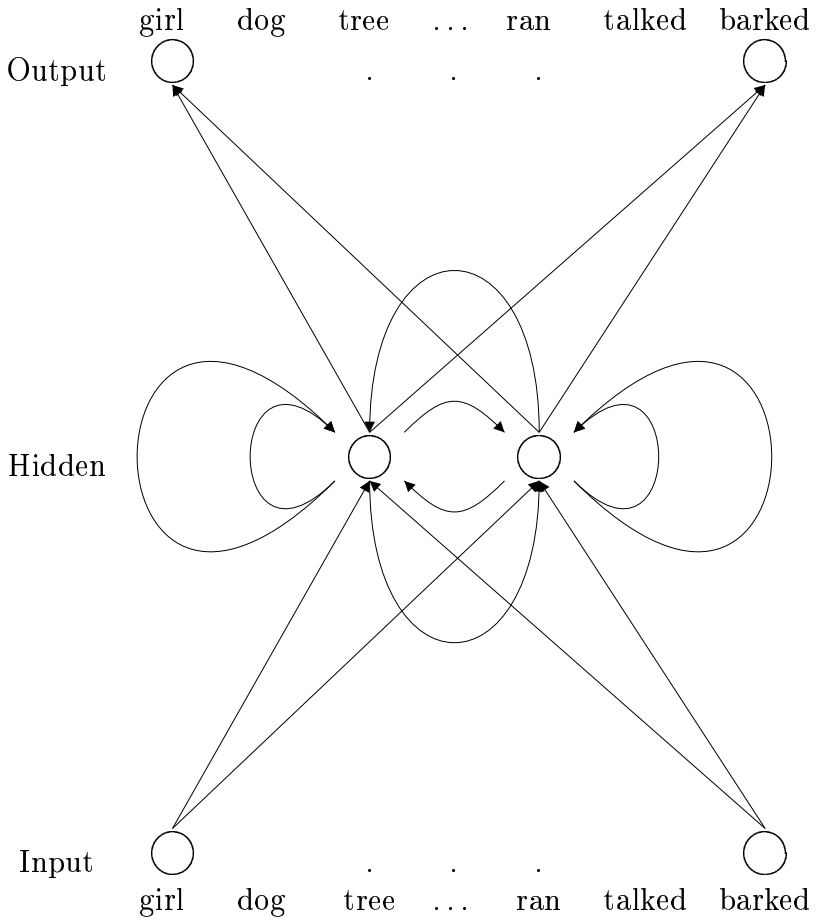

Figure 1: The Bramble Network (BRN).

so that

$$\frac{dE_{Ci}}{dw_{ij}} = 2\sigma_C(v_j)(net_i - v_i). \tag{5}$$

Equation (5) says, in effect, change the weights in the direction that minimizes the magnitude of recent activation change. Continuous weight learning is applied only when the network is close to a stable state (though crucially, not at it). It thus moves the stable state in the direction of the initial state, causing bifurcations when widely separated initial states are associated with a single attractor. The overall effect is that the attractors of the continuous weights tend to track the centers of masses of clusters defined by the discrete weights (Tabor et al., 1997). It is crucial to include the continuous space biases, b_i, and to tune these biases along with the other weights. Otherwise, the symmetry of the logistic function prevents effective tracking of cluster means.

We trained the network on the task of predicting the next word in a simple artificial language. Although it would be impossible for any mechanism to predict each successive word precisely without knowledge of the specific random character of the underlying process, if the process had some syntactic structure, the network would be able to pick up on that structure and constrain its guesses. In fact, in keeping with much recent work interpreting neural networks as statistical analyzers, prediction networks like the SRN and the BRN tend to distribute activation over the output units in proportion to the symbol emission probabilities associated with the current state of processing (Elman, 1990; Servan-Schreiber et al., 1991; Casey, 1996).[3] The details of the training procedures are described in the next section.

2.1 Case 1: Frequency Sensitivity

Sentence-structure seems to involve state-machine sequencing. There is much to recommend the view that the human language processor is best approximated by an infinite state device—e.g., (Chomsky, 1956)—we take up this issue under Case 2 below. But even within finite-state subdomains of natural language, there is a question of how categories can be learned and how

[3]We intend *state* in the sense of Crutchfield (Crutchfield, 1994): two instances in time belong to the same state if their expected futures are identical.

within-category variation can be efficiently handled.

One aspect of within-category variation is the inverse correlation between the frequency of a word and the amount of time it takes a language user to respond to it. This correlation has been demonstrated many times in isolated word reading studies. It also applies when words are read in a sentence context, where it is independent of the confound of word length (Inhoff and Rayner, 1986; Rayner and Duffy, 1986).

The aim in this study was to see if the Bramble Net would (i) learn the syntactic structure of a sentence generator from a sequence of examples of its grammatical sentences, (ii) develop a topology which would map, in a systematic way, onto the abstract structure of the generating process (Casey, 1996), and (iii) exhibit a correlation between frequency and processing time like that found in people.

To carefully study the BRN's response to frequency contrasts, we examined a grammar with a very simple phrasal organization: every sentence had the form (1).

(1) Noun Verb p

The Noun class contained 26 different words, the verb class contained 10 different words. Counting "p" (which stands for "period") there were 37 distinct words in lexicon of the language. Each of these words corresponded to a unique unit on the input layer of the BRN, whose activation value was clamped to 1 whenever its word was presented to the network.

The probability distributions characterizing transitions from the Nouns to the Verbs were designed to approximate the range of distributions seen in natural languages. A few Nouns were very high in frequency and tended to be followed by a small subset of verbs (these correspond to human-referring nouns in natural language corpora). A larger subset of the Nouns had somewhat lower frequency and tended to be used with the same set of verbs predominantly, but with a number of special verb preferences in individual cases (these correspond, approximately, to the many entities which we speak of anthropomorphically—animals, machines, companies, etc.). Finally, an even larger subset of Nouns had very low individual frequencies and tended to select verbs in a quite distinct way from the prototypical nouns (these correspond to words for inanimate entities that do not have much in common with people). In other words, the distributions are distributed approximately

spherically around a prototype, with noun frequency declining with distance from the prototype—we made them hyperbolically declining, following the evidence of Zipf (Zipf, 1943). A set of distributions with these characteristics was generated formulaically.

The verbs, on the other hand, only made a transition to one element ("p"), so they were all associated with a single next-word distribution. Because of the structure of the Noun-to-Verb transitions, the prior probabilities of Verbs, like the prior probabilities of Nouns, were distributed approximately hyperbolically.

Sentences were generated randomly as just described, and strung together end-to-end for presentation to a network (Elman, 1990). Thus, "p" also had a (unique) next-word distribution.

The prototypical network in this study had 37 input units, 10 hidden units, and 37 output units. The learning rate was 0.002 for both continuous and discrete weights. Momentum was set to 0.9 for the discrete weights and 0.0 for the continuous weights. The network had to process some sentences rapidly, with no time to settle between successive words, and some sentences more slowly, with 50 cycles of processing ($\Delta t = 0.05$) between successive words. The mixture of fast and slow was random in the ratio 7:3. The psychological motivation for this training scenario is that people hear speech at different rates on different occasions and they have to be able to handle the variation. The modeling motivation is that the rapid presentation makes successive words closer together in the network's memory and thus helps it learn longer dependencies; the slower presentation allows it to develop attractor structures which reveal some of the category structure of the generating process.

Eight networks differing only their random initial weight settings were trained. Each network was trained on the randomly generated

output of the grammar until the average Kulback-Leibler divergence error ($E = \sum_i t_i log t_i / o_i$, where t_i is the target for unit i and o_i is its activation) per word during training was less than 0.05. The target distributions were computed from the training grammar. The 0.05 level was achieved for each network by the time 500,000 word presentations had been made. The network was tested under fast presentation (no settling) and slow presentation (400 cycles of settling, which closely approximated convergence for almost every word). With fast processing, the average divergence error was 0.0163 (s.d. = 0.0036 across the means for the eight networks). With slow process-

ing, the average divergence error was 0.1633 (s.d. = 0.1527). These numbers can be helpfully compared to the minimum *divergence distances* between target distributions under the generating grammar. The divergence distance between two probability distributions $\mathbf{p}_1$ and $\mathbf{p}_2$ was is twice the divergence between $\mathbf{p}_1$ and $(\mathbf{p}_1+\mathbf{p}_2)/2$. If the nouns, verbs, and periods were treated as three clusters, respectively, then the minimum divergence distance between clusters (computed as the minimum divergence distance between points in the clusters) was 1.3863. If all words were treated individually, then the minimum divergence distance between distinct distributions was 0.0116. With this comparison removed, the minimum distance was 0.0345. Thus, it can be said that except for a few cases, fast processing was distingushing the lexical states, and slow processing was distinguishing what may be called the *grammatical states*, corresponding to the classes, Noun, Verb, and "p".

Even though fast processing produces more accurate word-prediction, slow processing is of interest as a model of human behavior in sentence processing. The convergence times provide an explicit model of reading times.

Rayner and Duffy (Rayner and Duffy, 1986) found that when people read sentences containing high and low frequency nouns, they read the high frequency nouns faster. We performed an analogous experiment on the networks.

After the networks were trained, we used the grammar to generate 600 words in sequence for each network. With the networks running in slow mode, we collected convergence times for each word. We determined when the network "arrived" at an attractor by asking when the distance between successive hidden unit states in the discrete approximation to continuity dropped below 0.005. This number was chosen so that the words in the corpus showed a significant range of convergence times. We also used the grammar to compute frequencies of the individual words. For the set of all 600 words in the corpus, the correlation between frequency and convergence time was significantly negative in all but one case (With the outlier—p = .19—removed, mean $R^2 = 0.37$; s.d. = 0.27; max $p < .0001$). For the set all the Nouns in each corpus, the correlation was significantly negative in every case (mean $R^2 = 0.64$; s.d. = 0.21; max $p < .0001$).

Why does this frequency correlation obtain? Figure 2 shows a reduced-dimension plot of a sample of 600 trajectories associated with grammatical processing. The plot was generated by performing Principal Component Analysis (Jolliffe, 1986) on the set of all the points in the 600 trajectories

and plotting the projections of the trajectories on the subspace formed by the first two components. Each trajectory starts at the unmarked end of a curve and ends at a circle. The figure suggests that the network has formed three distinct manifolds (or, possibly, three separated segments of a single manifold) which attract the system's state. Each manifold (segment) corresponds to a grammatical class (Noun, Verb, or p). This classification was determined by checking the lexical labels of all the points in the sample. Consider the trajectories associated with the Nouns. The initial points of the trajectories have a geometry which approximates the geometry of the transition probability distributions in the output space: approximately spherical, with the highest frequency nouns in the center and the lower frequency nouns further out. The initial points have this structure because the accuracy training of the discrete weights creates a topography which maps, in a linearly separable fashion, onto the output distributions. Moreover, the attractive manifold of fixed points lies in the center of the sphere, relatively close to the initial points of the highest-frequency nouns. The manifold is located near the high-frequency cases because the stability training pulls it most strongly in the direction of the highest-frequency nouns.

One may ask why there are three manifold segments rather than many. After all, each individual noun has a distinct future in the underlying process, and those distinctions are encoded in the initial states. Why should the stable states not divide into many disconnected manifolds to reflect those differences? It is possible that, with sufficiently long training, they will, but we have not seen it yet. We tried one run up to 4 million pattern presentations. The manifold structure still looked stable in a PCA representation, although the Noun section had stretched considerably relative to the Verb and "p" manifolds. A relationship between the degree of asymmetry in an initial state distribution and the representational power of the continuous weights must determine which subsets of points get mapped to connected manifolds and which get mapped to separate manifolds. Characterizing this relationship may be analogous to answering the question, What are the abstract grammatical categories that organize the structure of a language? One appealing feature of the current model is that it provides a formal framework for addressing this question. Hypothesized abstract grammatical classes have been very helpful in the development of linguistic understanding—they are among the most successful "mental entities" in cognitive psychology. But their definitions depend on a set of grammatical "diagnostics" whose membership is

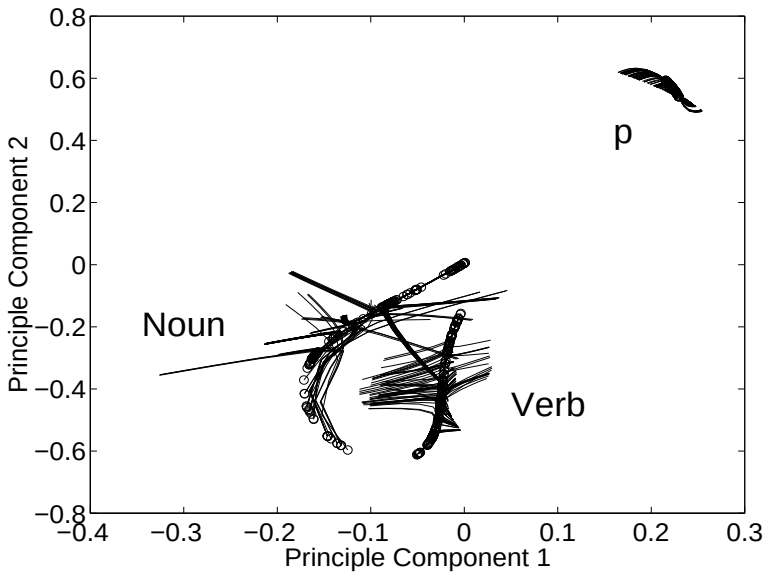

Figure 2: Reduced-dimension plot of the manifolds associated with distinct lexical classes.

controversial, and not completely specified. A formal, learning-based definition of categories would strengthen the field considerably.

The current study may leave one wondering what point there is in having the discrete recurrent connections in the BRN. After all, the dependencies of the "Noun Verb p" grammar could be learned by a feedforward network. In other experiments, we have studied the performance of the BRN and the related VSG model on tasks involving context-dependent interpretation of words (Tabor and Tanenhaus, 1999; Tabor and Hutchins, 2000). One discovery of such studies is that difficulty on one word often carries over to difficulty on succeeding words (see also Case 2 below). This phenomenon is an instance of *representational inertia*, the persistence over time of mental representations in the face of contradictory evidence. The phenomenon has been observed empirically in many reading studies (Rayner and Duffy, 1986; Rayner, 1998). In fact, the current BRN exhibits a tendency toward representational inertia as well, even though there is never any need to carry information forward more than one word at a time. One of the eight networks tends, in slow processing, to mistake Verbs for Nouns, resulting in high processing times on the Verbs. If representational inertia is involved here, then the high reading times should persist into the next state, independently of that state's prior tendencies. Indeed, the correlation between convergence time on "p" (the word that always follows Verb) and convergence time on the word preceding "p" (always Verb) was highly significant for this network ($R^2 = 0.89$, $p < .0001$).

2.2 Case 2: Phrase-structure

Although the results of the previous section are encouraging, merely handling frequency sensitivity is not a convincing demonstraton that dynamical recurrent networks are up to the job of characterizing the rich structure that seems to exist in human languages. Phrase structure, on the other hand, is a powerful organizing mechanism which greatly simplifies the description of natural language syntax (Chomsky, 1957). Here, we describe a class of dynamical computers, called Dynamical Automata (DAs) (Tabor, 1998; Tabor, 2000), which can recognize and generate phrase structure languages, as well as many other complex languages.

In the general case, phrase structure requires an unbounded stack. But people's ability to store syntactic stack states seems to taper off around 3

or 4 levels of embedding (e.g., (Miller and Chomsky, 1963; Christiansen and Chater, 1999)). What mechanism can capture both the simplicity fact and the tapering capacity fact? A pushdown automaton with a limit on the length of its stack does a reasonable job of approximating the data (Marcus, 1980), but the assumption of a hard cutoff is dubious because of the variation with context of a single person's ability to process center embedded structures (Schlesinger, 1968). It would be nice to motivate the limit on memory, too. DAs do simple stack-computation under ideal conditions but suffer human-like forgetting in a world with noise. Thus, DAs provide a plausible method of encoding stacks in a distributed, neuron-like representation. (See also Chapter 5 of this volume).

We will give an example of a DA for a simple context free grammar, summarize a theorem about formal equivalence to context free grammars, and then show how a DA model of natural language predicts memory load effects.

2.2.1 An example Dynamical Automaton

Under one definition, a fractal is a set of points which is self-similar at arbitrarily small scales (see also Chapter **??** of this volume). Figure 3 shows a diagram of the fractal called the Sierpinski Triangle (the letter labels in the diagram will be explained presently). The Sierpinski triangle, a kind of Cantor set, is the limit of the process of successively removing the "middle quarter" of a triangle to produce three new triangles.

The grammar shown in Table 1 is a context free grammar which includes some center-embedded structures and thus cannot be emulated by a finite state machine. A pushdown automaton for the language of Grammar 2 would need to keep track of each "abcd" string that has been started but not completed. For this purpose it could store a symbol corresponding to the last letter of any partially completed string on a pushdown stack. For example, if it stored the symbol "A" whenever an embedding occurred under "a", "B" for an embedding under "b" and "C" for an embedding under "c", the stack states would be members of {A, B, C}*. We can use the Sierpinski Triangle to keep track of the stack states for Grammar 2. Consider the labeled triangle in Figure 3(a). Note that all the labels are at the midpoints of hypotenuses of subtriangles (e.g., the label "CB" corresponds to the point, (0.125, 0.625)). The labeling scheme is organized so that each member of {A, B, C}* is the

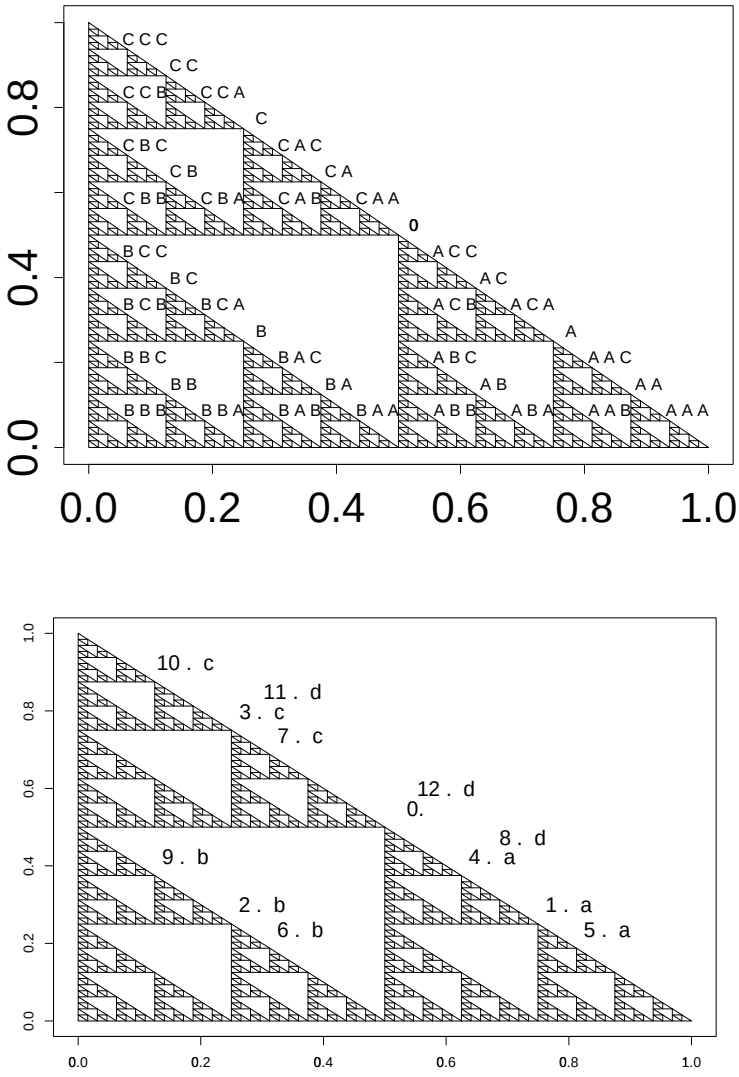

Figure 3: (a) Sierpinski triangle with stack states labelled. (b) A sample trajectory.

label of some midpoint (only stacks of cardinality 3 are shown).

Rule 1a.	S $\rightarrow$ A B C D
Rule 1b.	S $\rightarrow \epsilon$
Rule 2a.	A $\rightarrow$ a S
Rule 2b.	A $\rightarrow$ a
Rule 3a.	B $\rightarrow$ b S
Rule 3b.	B $\rightarrow$ b
Rule 4a.	C $\rightarrow$ c S
Rule 4b.	C $\rightarrow$ c
Rule 5a.	D $\rightarrow$ d S
Rule 5b.	D $\rightarrow$ d

Table 1: Grammar 2. (Implemented in Dynamical Automaton 1).

We define a DA (called DA 1) that recognizes the language of Grammar 2 by the Input Map shown in Table 2. The essence of the DA is a two-element vector, $\mathbf{z}$, corresponding to a position on the Sierpinski triangle. The DA functions as follows: when $\mathbf{z}$ is in the subset of the plane specified in the "Compartment" column, the possible inputs are those shown in the "Input" column. Given a compartment and a legal input for that compartment, the change in $\mathbf{z}$ that results from reading the input is shown in the "State Change" column. If we specify that the DA must start with $\mathbf{z} = (1/2, 1/2)$, make state changes according to the rules in Table 2 as symbols are read from an input string, and return to $\mathbf{z} = (1/2, 1/2)$ (the Final Region) when the last symbol is read, then the computer functions as a recognizer for the language of Grammar 2. To see this intuitively, note that any subsequence of the form "a b c d" invokes the identity map on $\mathbf{z}$. Thus DA 1 is equivalent to the nested finite-state machine version of Grammar 2. For illustration, the trajectory corresponding to the string "a b c a a b c d b c d d" is shown in Figure 3(b) (1.a is the position after the first symbol, an "a", has

been processed; 2.b is the position after the second symbol, a "b" has been processed, etc.)

Region	Input	State Change
$z1 > 1/2$ and $z2 < 1/2$	b	$\mathbf{z} \leftarrow \mathbf{z}$ - (1/2, 0)
$z1 < 1/2$ and $z2 < 1/2$	c	$\mathbf{z} \leftarrow \mathbf{z} + (0, 1/2)$
$z1 < 1/2$ and $z2 > 1/2$	d	$\mathbf{z} \leftarrow 2(\mathbf{z}$ - (0, 1/2))
Any	a	$\mathbf{z} \leftarrow 1/2\ \mathbf{z} + (1/2, 0)$

Table 2: Dynamical Automaton 1.

The computations of this Dynamical Automaton bear a close resemblance to the empirically observed computations of Simple Recurrent Networks (SRNs). Rodriguez, et al., (Rodriguez et al., 1999) found that SRNs trained on the language $a^n b^n$ performed their computations on a geometric series fractal. Elman (Elman, 1991) examined the many-dimensional hidden unit space of an SRN trained on more elaborate recursive languages and found that different lexical classes corresponded to different subregions of the space. Likewise, in the example above, the three lexical classes, A, B, and C correspond to three distinct regions of the representation space (each class has only one member). The item "d" does not need a class of its own because its occurrence always puts the computer into a state corresponding to one of the other three classes. Elman also noted that the SRN followed similarly-shaped trajectories from region to region whenever it was processing a phrase of a particular type, with slight displacements differentiating successive levels of embedding. Here, the single phrase S is also associated with a characteristic (triangular) trajectory wherever it occurs and slight displacements also differentiate successive levels of embedding.

2.2.2 Formal relationships between dynamical automata and symbolic computers

One can construct a wide variety of computing devices that organize their computations around fractals. At the heart of each fractal computer is a set of iterating functions that have associated stable states and which can be analyzed using tools of dynamical systems theory (Barnsley, 1993). One species

of Dynamical Automata, called Pushdown Dynamical Automata (PDDAs) is a class of generator/recognizers for CFLs. DA 1 above is an example of a PDDA. Tabor (Tabor, 2000) formalizes the equivalence. An appealing consequence of the analysis is that various familiar symbolic computing devices (e.g. context free grammars, context-sensitive grammars, queue-based grammars. etc. see (Moore, 1998)) can be identified in metric spaces where they bear distance relationships to other computing devices. The (geo)metric perspective on these relationships may be useful in the problem of navigating among models via learning (Tabor, 1998; Tabor, 2000).

2.2.3 Simulation of memory load: Multiply center-embedded constructions.

Several researchers (e.g., (Miller and Chomsky, 1963; Bach et al., 1986; Christiansen and Chater, 1999)) have emphasized that exact computation of arbitrary center-embedding structures is not very human-like. In fact, people seem usually to be able to comprehend at most three layers of clausal structure and their comprehension of three-layer structures (e.g., "The squirrel the dog Jane owned chased escaped.") is intermittent. The Dynamical Automata described in the previous section perform perfectly at all levels of embedding, and thus are not particularly human-like as such. However, it is natural to assume that some noise distorts the computation of activation values. With fixed-variance Gaussian noise added to each activation value and a tolerance of one-standard-deviation permitted for the final state, the performance of the network described above degrades in a realistic way with growth in the number of center-embeddings (Figure 4). This account is appealing in comparison to proposals to explicitly limit the storage capacity of a symbolic parser (Lewis, 1996; Gibson, 1998) because it links processing failure to a plausible property of neurons. Casey (Casey, 1996) notes that a recurrent network with bounded state space and measurable indeterminacy in its states cannot recognize an infinite-state language. The current perspective emphasizes the importance of not treating Casey's point as an argument for studying only finite-state models of cognition: the exact computation provides a useful characterization of the organizing system underlying the noisy computation. The distinction between noisy and exact is thus similar to Chomsky's (Chomsky, 1957) distinction between performance and competence.

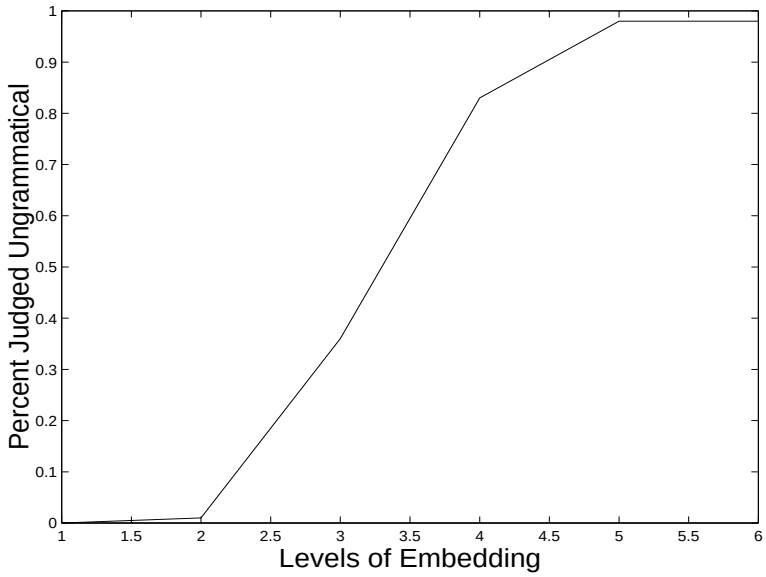

Figure 4: The performance of the network described above degrades in a realistic way with growth in the number of center-embeddings.

2.2.4 Subject versus object relative clauses.

King and Just (King and Just, 1991) and Gibson and Ko (Gibson and Ko, 1999) measured reading times phrase by phrase in subject-extracted relative clauses ("subject relatives") like (2a) and object-extracted relative clauses ("object relatives") like (2b).

(2a) The girl who met the boy liked fish.

(2b) The girl who the boy met liked fish.

They found that the highest mean reading times were higher in object-relatives than in subject relatives.

Gibson (Gibson, 1998) proposes a model of these and many other reading time phenomena based on the notion that the sentence processor incurs a load when it has to integrate temporally separated events. For example, it is costly to integrate the word "met" with the word "who" (its grammatical object) in (1b) because it is separated from "who" by the phrase "the boy". Furthermore, it is more costly to integrate "met" with "who" in (1b) than it is in (1a) because "met" is further away from "who" in (1b) than (1a). Gibson's model gives a close fit to the reading time profiles across words in the sentence (Figure 5(a)), although it puts the maximal integration costs at the same value in both sentences, contra the findings of King and Just.

The noisy Dynamical Automaton model assumes that a small amount of Gaussian noise (variance = 0.02 units in the metric space of the fractal) distorts the representation every time a new word is read. The system recovers from the noise by moving back (along a straight line in its representation space) toward the point it is supposed to be at until it comes within a particular small radius of that point (tolerance radius = 0.02). Predicted reading time is proportional to the distance the system moves to get within tolerance. This mechanism takes its inspiration from dynamical models of parsing like that of Case 1 above (see also (Tabor et al., 1997; McRae et al., 1998)) in which the parser, upon processing each word, falls into a basin of attraction corresponding to the (presumed) correct parse. Actual dynamical implementation of the correction mechanism is a focus of current research.

The Input Map for the Dynamical Automaton we used to model transitive sentences with relative clause modifiers is shown in Table 3. The automaton uses 9 partition states and moves around on a 3-dimensional fractal. Table 4

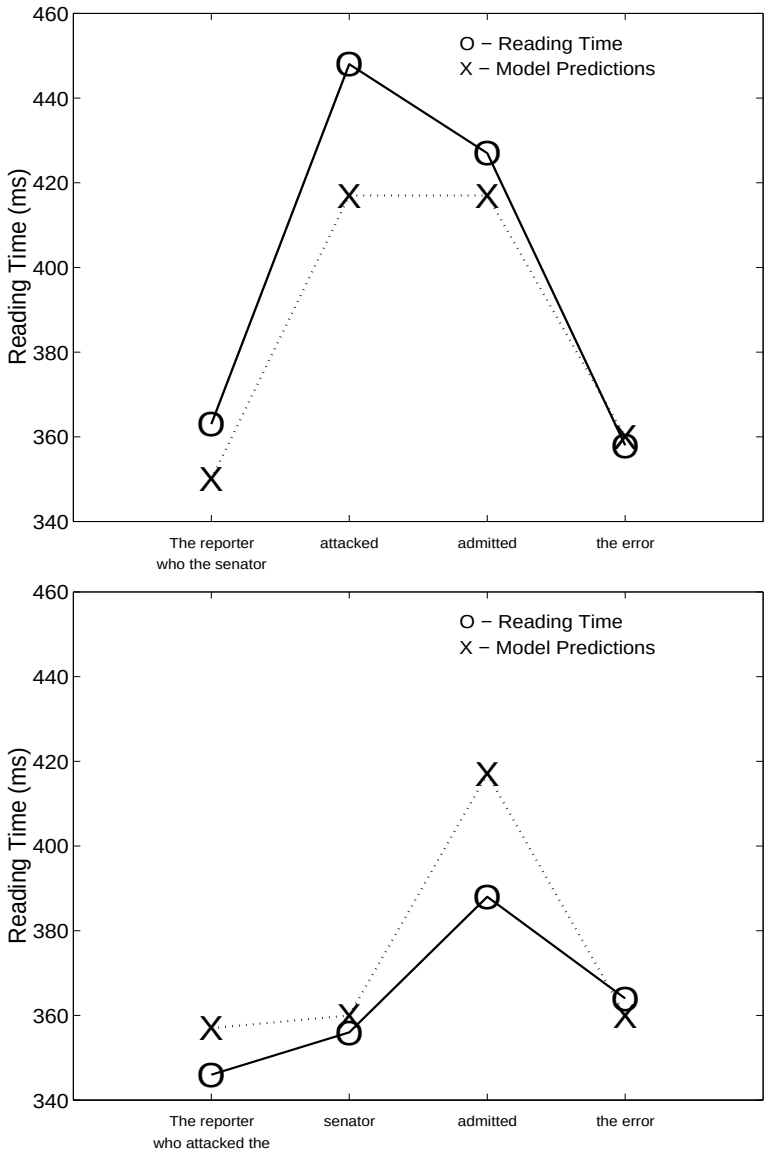

Figure 5: Comparison of Gibson and Ko's human reading time data with the predictions of Gibson's integration cost model.

gives the compartment definitions. The automaton works as follows. Each transition from word to word within a phrase involves moving between regions of the fractal representing different parse states. If some constituent must be remembered, a vector symbol is "pushed" onto the neural stack in a similar manner to the way the symbols A, B, and C were pushed onto the stack in the previous example. The symbol is later removed from the stack when the continuation of its constituent is encountered.[4]

Figures 6(a) and 6(b) compare the predictions of the Dynamical Automaton model to the results of Gibson and Ko (Gibson and Ko, 1999). The model's time predictions (proportional to the distance it travels to get within tolerance) are scaled by an affine transformation to match the human data in mean and range across all data points. The profile fit is, as with Gibson's model, quite a close one. The model predicts high reading times at the points where information that has been pushed onto the stack is being recovered. The reason for the high reading times at these points is that the fixed-variance noise has a stronger effect on information that is encoded at a smaller scale. For example, the information about the subject ("girl") has been scaled by a factor of 1/2 during the processing of the phrase "who met boy" in the Subject Relative case.

The model predicts the observed contrast between the maximal reading times of Subject and Object Relatives. Although Gibson and Ko don't report on the significance of this contrast in their data, King and Just found very similar profiles and observed a significant difference between the reading times at the second verb in the two cases—"likes" in the example ($F(1, 32) = 23.99$, $p < .001$). The Dynamical Automaton model predicts this contrast because of its assumption that recovery can be only partial (within the tolerance of 0.02 units in hidden unit space). Thus, difficulty with integration at "met" can carry over to the next word. In Subject Relatives (1b), this representational inertia has no prominent effect on reading times because "met" is relatively easy to process and the next word ("boy") is also easy to process.

[4]Consistent with psycholinguistic work on filled-gap effects (Stowe et al., 1991), the model aggressively posits traces whenever a licensing verb comes along (i.e., it does not wait until the next word provides evidence that a missing constituent has been passed). This assumption has the consequence that integration of the arguments of a verb (e.g., "who", "the boy") occurs when the verb ("met") is read. On the other hand, consistent with the tendency toward Late Closure (e.g., (Frazier and Rayner, 1982)) constituent closure is not performed until the following word provides evidence for closure.

Compartment	Input	Transition	Symbolic Meaning
Start	N	$\mathbf{z} \leftarrow 1/2\ \mathbf{z} + (1/2, 0, 1/2)$	push NSubj1
Comp	V	$\mathbf{z} \leftarrow \mathbf{z}$ - (1/2, 0, 0)	switch(V, Comp)
	N	$\mathbf{z} \leftarrow 1/2\ \mathbf{z} + (1/2, 1/2, 1/2)$	push NSubj2
V	N	$\mathbf{z} \leftarrow \mathbf{z} + (-1/2, 0, 1/2)$	switch(V, NObj)
NSubj1	Comp	$\mathbf{z} \leftarrow 1/2z + (0, 1/2, 0)$	push Comp2
	V	$\mathbf{z} \leftarrow \mathbf{z} + (0, 0, 1/2)$	switch(NSubj, V)
NSubj2	Comp	$\mathbf{z} \leftarrow 1/2\ \mathbf{z}$	push Comp2
	V	$\mathbf{z} \leftarrow 2(z - (1/2, 1/2, 1/2)) + (0, 0, 1/2)$	pop NSubj2 then switch(Comp, Nobj)
NObj1	Comp	$\mathbf{z} \leftarrow \mathbf{z} + (0, 0, -1/2)$	switch(NObj, Comp)
	V	$\mathbf{z} \leftarrow 2[2(z - (0, 0, 1/2)) - (1/2, 1/2, 1/2)] + (0, 0, 1/2)$	pop NObj1 then pop NSubj2 then switch(Comp, NObj)
	End	$\mathbf{z} \leftarrow 2(\ \mathbf{z} - (0, 0, 1/2))$	pop NObj1
NObj2	Comp	$\mathbf{z} \leftarrow \mathbf{z}$ - (0, 0, 1/2)	switch(NObj, Comp)
	V	$\mathbf{z} \leftarrow 2(\ \mathbf{z} - (0, 1/2, 1/2)) - (0, 0, 1/2)$	pop NObj2 then switch(NSubj, V)

Note: switch(X, Y) means switch from control state X to control state Y.

Table 3: Dynamical Automaton 2: transitions.

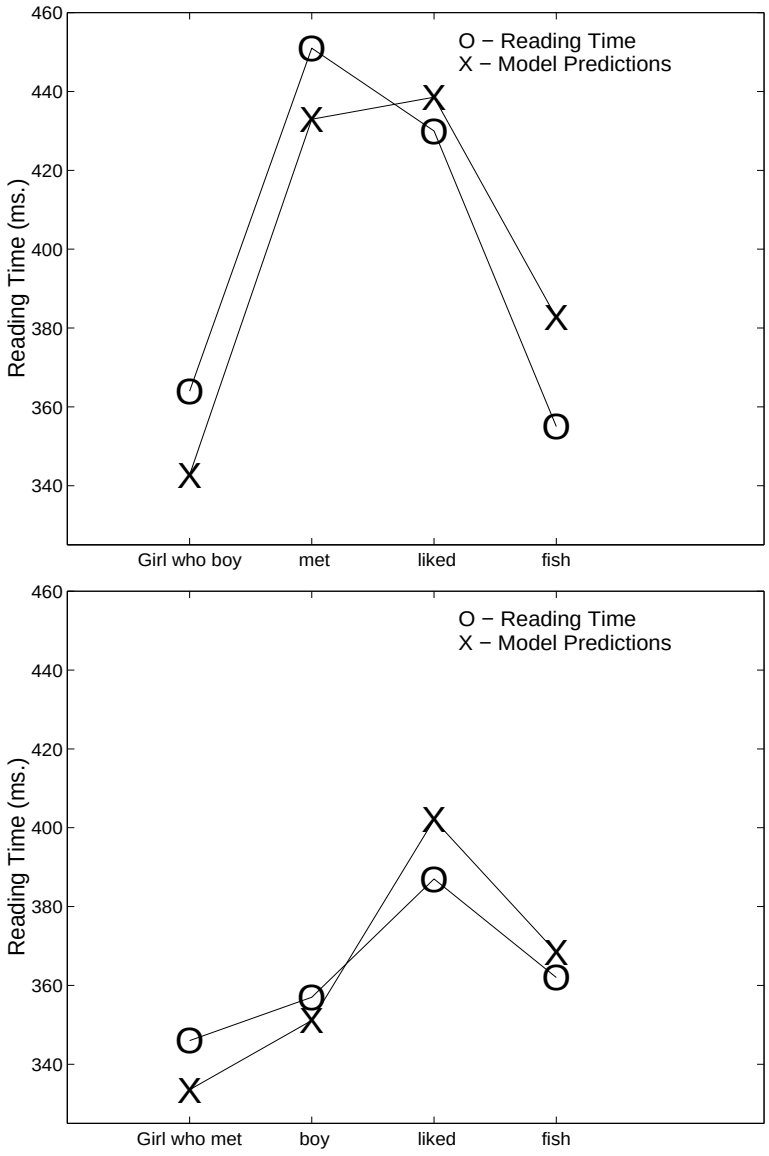

Figure 6: Comparison of Gibson and Ko's human reading time data with the predictions of DA 2.

Name	Definition
Start	(1/2, 1/2, 1/2)
Comp1	(0, 0, 0) + **opencube**
Comp2	(0, 1/2, 0) + **opencube**
V1	(1/2, 0, 0) + **opencube**
V2	(1/2, 1/2, 0) + **opencube**
NObj1	(0, 0, 1/2) + **opencube**
NObj2	(0, 1/2, 1/2) + **opencube**
NSubj1	(1/2, 0, 1/2) + **opencube**
NSubj2	(1/2, 1/2, 1/2) + **opencube**

Note: **opencube** is the set $\{(x, y, z) : 0 < x < 1/2, 0 < y < 1/2, 0 < z < 1/2\}$.
Note: Compartment A as labelled in Table 3 is the union of the compartments A1 and A2 shown above for A $\in \{Comp, V, NObj, NSubj\}$.

Table 4: Dynamical Automaton 2: compartment definitions.

But in Object Relatives (1a) "met" is especially hard to process and this difficulty carries over onto the word "likes", which is hard in the first place. Gibson's account does not make this prediction because the load imposed by integration does not interact dynamically with the processor: the load is computed at a word and has its entire effect at that word, so difficulty on one word cannot carry over to a following one.

3 Conclusions

We have described several models of human sentence processing which use the constructs of dynamical systems theory. The models use a mixture of discrete and continuous state change. They compute an initial response to each successive word discretely, and then undergo a continuous adjustment process which clarifies the syntactic classification of the word. Dynamical constructs help in several ways. The dynamics provide an explicit model of the time course of processing, something which can be easily and objectively measured. The models' metric properties account for observed fine-grained

differences between similar items: metric contrasts underlie the frequency effects in Case 1 and the inertia effects in Cases 1 and 2. The models' topological properties provide insight into the structural principles which organize their computations. Topological contrasts define the lexical class manifolds in Case 1 and the fractal organization of phrasal embedding in Case 2.

Where does one go from here? This work suggests that a fundamental challenge for cognitive science is to properly characterize the relationship between the metric and topological organization of systems. The current studies achieve this integration partially, but not wholly. For example, the Bramble Network is good at learning, handles noise well, and induces structures that resemble rudimentary structural features of natural languages. But it struggles to learn structures of the complexity that Dynamical Automata can handle. On the other hand, although Dynamical Automata can perform computations of arbitrary complexity (Moore, 1998), we do not yet know how their weights can be learned from data, and their operation in noise needs to be more fully defined and explored. Thus it is desirable to better integrate the Bramble Net and Dynamical Automaton perspectives.

In sum, dynamical recurrent networks provide new insight because of the link they make between metric and topological properties of complex systems. The topological structures of dynamical recurrent networks may provide a basis for formal models of the mental entities which cognitivist theories of mind propose. Their metric properties provide a suitable environment for learning, and allow the networks to make testable quantitative predictions. Thus, the link may strengthen the mathematical and empirical grounding of the cognitive sciences, and may, conversely, help reductionist approaches to complex systems (e.g., physics, chemistry, neurobiology) identify useful systems-level generalizations.

References

Bach, E., Brown, C., and Marslen-Wilson, W. (1986). Crossed and nested dependencies in german and dutch: A psycholinguistic study. *Language and Cognitive Processes*, 1:249–262.

Barnsley, M. ([1988]1993). *Fractals Everywhere, 2nd ed.* Academic Press, Boston.

Bergener, T., Bruckhoff, C., Dahm, P., Janben, H., Joublin, F., Menzner, R., Steinhage, A., , and Von Seelen, W. (1999). Complex behavior by means of dynamical systems for an anthropomorphic robot. *Neural Networks*, 12(7 & 8).

Burgess, C. and Lund, K. (1997). Modeling parsing constraints with a high-dimensional context space. *Language and Cognitive Processes*, 12(2/3):177–210.

Casey, M. (1996). The dynamics of discrete-time computation, with application to recurrent neural networks and finite state machine extraction. *Neural Computation*, 8(6).

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124. A corrected version appears in Luce, Bush, and Galanter, eds., 1965 *Readings in Mathematical Psychology, Vol. 2*.

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

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

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

Cottrell, G. W. and Small, S. (1984). Viewing parsing as word sense discrimination: A connectionist approach. In Bara, B. and Guida, G., editors, *Computational Models of Natural Language Processing*, pages 91–119. North Holland, Amsterdam.

Crutchfield, J. P. (1994). The calculi of emergence: Computation, dynamics, and induction. *Physica D*, 75:11–54. In the special issue on the Proceedings of the Oji International Seminar, *Complex Systems—from Complex Dynamics to Artificial Reality*.

Elman, J. (1995). Language as a dynamical system. In Port, R. and van Gelder, T., editors, *Mind as Motion: Explorations in the Dynamics of Cognition*. MIT Press, Cambridge, MA.

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

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

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

Gibson, E. (1998). Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Gibson, E. and Ko, K. (1999). Processing main and embedded clauses. Manuscript, Department of Brain and Cognitive Sciences, MIT.

Inhoff, A. W. and Rayner, K. (1986). Parafoveal word processing during eye fixations in reading: Effects of word frequency. *Perception and Psychophysics*, 40:431–439.

Jolliffe, I. T. (1986). *Principal component analysis*. Springer-Verlag, New York.

Jordan, M. I. and Rumelhart, D. E. (1992). Forward models: supervised learning with a distal teacher. *Cognitive Science*, 16.

Juliano, C. and Tanenhaus, M. K. (1994). A constraint-based lexicalist account of the subject/object attachment preference. *Journal of Psycholinguistic Research*, 23(6):459–471.

Kempen, G. and Vosse, T. (1989). Incremental syntactic tree formation in human sentence processing: a cognitive architecture based on activation decay and simulated annealing. *Connection Science*, 1:273–290.

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

Kremer, S. C. (1996). A theory of grammatical induction in the connectionist paradigm. PhD Thesis, Department of Computing Science, Edmonton, Alberta.

Landauer, T. K. and Dumais, S. T. (1997). A solution to Plato's problem: The latent semanatic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240.

Large, E. and Kolen, J. (1999). Resonance and the perception of musical meter. In Griffith, N. and Todd, P. M., editors, *Musical Networks*, pages 65–96. MIT Press, Cambridge, MA.

Lewis, R. (1996). Interference in short-term memory: The magical number two (or three) in sentence processing. *Journal of Psycholinguistic Research*, 25(1).

Marcus, M. (1980). *A Theory of Syntactic Recognition for Natural Language*. MIT Press, Cambridge, Massachusetts.

McClelland, J. L. and Kawamoto, A. H. (1986). Mechanisms of sentence processing: Assigning roles to constituents. In McClelland, J. L. and Rumelhart, D. E., editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, pages 272–326. MIT Press, Cambridge, MA.

McRae, K., Spivey-Knowlton, M. J., and Tanenhaus, M. K. (1998). Modeling the influence of thematic fit (and other constraints) in online sentence comprehension. *Journal of Memory and Language*, 38:283–312.

Miller, G. and Chomsky, N. (1963). Finitary models of language users. In Luce, D. R., Bush, R. R., and Galanter, E., editors, *Handbook of Mathematical Psychology, Vol. II*. John Wiley, New York.

Moore, C. (1998). Dynamical recognizers: Real-time language recognition by analog computers. *Theoretical Computer Science*, 201:99–136.

Page, M. P. A. (1999). Modelling the perception of musical sequences with self-organizing neural networks. In Griffith, N. and Todd, P. M., editors, *Musical Networks*, pages 65–96. MIT Press, Cambridge, MA.

Pearlmutter, B. A. (1989). Learning state space trajectories in recurrent neural networks. *Neural Computation*, 1:263–269.

Pearlmutter, B. A. (1995). Gradient calculations for dynamic recurrent neural networks: A survey. *IEEE Transactions on Neural Networks*, 6(5):1212–1228.

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

Pollack, J. B. (1990). Recursive distributed representations. *Artificial Intelligence*, 46:77–106. Special issue on Connectionist symbol processing edited by G. E. Hinton.

Port, R. and van Gelder, T., editors (1995). *Mind as Motion: Explorations in the Dynamics of Cognition*. MIT Press, Cambridge, MA.

Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124(3):372–422.

Rayner, K. and Duffy, S. A. (1986). Lexical complexity and fixation times in reading: Effects of word frequency, verb complexity, and lexical ambiguity. *Memory and Cognition*, 14:191–201.

Rodriguez, P., Wiles, J., and Elman, J. (1999). A recurrent neural network that learns to count. *Connection Science*, 11(1):5–40.

Rohde, D. and Plaut, D. (1999). Language acquisition in the absence of explicit negative evidence: How important is starting small? *Journal of Memory and Language*, 72:67–109.

Rueckl, J. (1995). Ambiguity and connectionist networks: Still settling into a solution: Comment on joordens and besner (1994). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 21:501–508.

Rumelhart, D., Durbin, R., Golden, R., and Chauvin, Y. (1995). Backpropagation: The basic theory. In *Backpropagation: Theory, Architectures, and Applications*. Lawrence Erlbaum Associates.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E., McClelland, J. L., and the PDP Research Group, editors, *Parallel Distributed Processing, Volume I*, pages 318–362. MIT Press.

Schlesinger, I. M. (1968). *Sentence Structure and the Reading Process.* The Hague, Mouton.

Schöner, G., Dose, M., and Engels, C. (1995). Dynamics of behavior: theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems*, 16:213–245.

Seidenberg, M. S. and McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review*, 96:447–452.

Selman, B. and Hirst, G. (1985). A rule-based connectionist parsing system. In *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, pages 212–221. .

Servan-Schreiber, D., Cleeremans, A., and McClelland, J. L. (1991). Graded state machines: The representation of temporal contingencies in simple recurrent networks. *Machine Learning*, 7:161–193.

Siegelmann, H. (1996). The simple dynamics of super Turing theories. *Theoretical Computer Science*, 168:461–472.

Siegelmann, H. T. and Sontag, E. D. (1991). Turing computability with neural nets. *Applied Mathematics Letters*, 4(6):77–80.

Stowe, L. A., Tanenhaus, M. K., and Carlson, G. M. (1991). Filling gaps on-line: Use of lexical and semantic information in sentence processing. *Language and Speech*, 34:319–340.

Tabor, W. (1998). Dynamical automata. Technical Report No. TR98-1694, Cornell Computer Science Department.

Tabor, W. (2000). Fractal encoding of context-free grammars in connectionist networks. *Expert Systems: The International Journal of Knowledge Engineering and Neural Networks*, 17(1):41–56.

Tabor, W. and Hutchins, S. (2000). Mapping the syntax/semantics coastline. In *Proceedings of the 22nd Annual Meeting of the Cognitive Science Society*. Lawrence Erlbaum Associates, Mahwah, NJ.

Tabor, W., Juliano, C., and Tanenhaus, M. (1997). Parsing in a dynamical system: An attractor-based account of the interaction of lexical and structural constraints in sentence processing. *Language and Cognitive Processes*, 12(2/3):211–271.

Tabor, W. and Tanenhaus, M. K. (1999). Dynamical models of sentence processing. *Cognitive Science*, 23(4):491–515.

Tani, J. (1998). An interpretation of the 'self' from the dynamical systems perspective: a constructivist approach. *Journal of Consciousness Studies*, 5(5-6).

Tani, J. and Nolfi, S. (1999). Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems. *Neural Networks*, 12:7 & 8.

van Gelder, T. (1998). The dynamical hypothesis in cognitive science. *Behavioral and Brain Sciences*, 21:615–665.

Vosse, T. and Kempen, G. (1999). Syntactic structure assembly in human parsing: a computational model based on competitive inhibition and a lexicalist grammar. Submitted manuscript, Department of Psychology, Leiden University.

Williams, R. J. and Peng, J. (1990). An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural Computation*, 2:490–501.

Zipf, G. K. (1943). *Human Behavior and the Principle of Least Effort*. Hafner [1965].