NEURO-FRACTAL COMPOSITION OF MEANING: TOWARD A COLLAGE THEOREM FOR LANGUAGE

Simon D. Levy [1]
levys@wlu.edu
Computer Science Department
Washington & Lee University, Lexington, VA 24450, USA

ABSTRACT

This paper presents languages and images as sharing the fundamental property of self-similarity. The self-similarity of images, especially those of objects in the natural world (leaves, clouds, galaxies), has been described by mathematicians like Mandelbrot, and has been used as the basis for fractal image compression algorithms by Barnsley and others. Self-similarity in language appears in the guise of stories within stories, or sentences within sentences (”I know what I know”), and has been represented in the form of recursive grammar rules by Chomsky and his followers. Having observed this common property of language and images, we present a formal mathematical model for putting together words and phrases, based on the iterated function system (IFS) method used in fractal image compression. Building (literally) on vector-space representations of word meaning from contemporary cognitive science research, we show how the meaning of phrases and sentences can likewise be represented as points in a vector space of arbitrary dimension. As in fractal image compression, the key is to find a set of (linear or non-linear) transforms that map the vector space into itself in a useful way. We conclude by describing some advantages of such continuous-valued representations of meaning, and potential implications.

NOMENCLATURE

Self-similarity, fractals, language, grammars, iterated function systems, recurrent neural networks.

INTRODUCTION: SELF-SIMILARITY

Self-similarity is a property by which an object contains smaller copies of itself at arbitrary scales. As noted by Mandelbrot (1988), this property is ubiquitous in the natural world, appearing in objects as diverse as leaves, mountain ranges, galaxies, and clouds. Figure 1 shows a mathematical abstraction of this concept, in which the original image is arrived at through a sequence of zooming operations.[2]

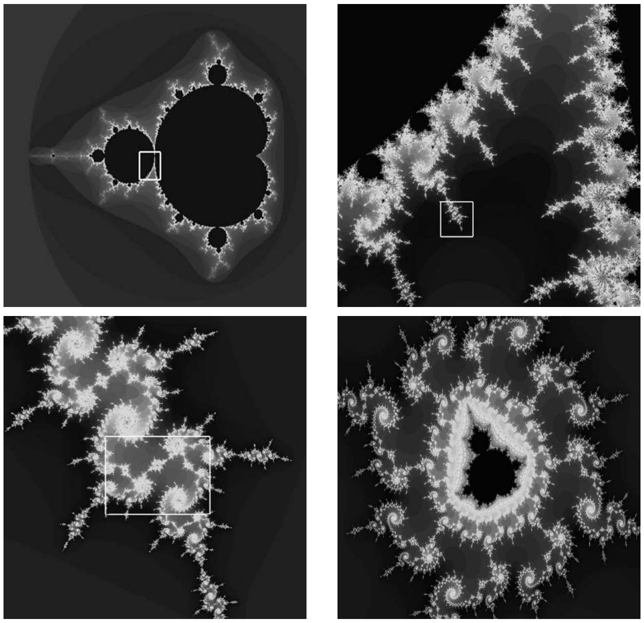

Figure 1: The Mandelbrot Set at various scales.

Self-similarity is also a hallmark of human language. The following verse, from a poem by Wallace Stevens (Stevens, Kermode, and Richardson 1997), meanders through a sequence of relative clauses, arriving ultimately at self-description:

I know noble accents
And lucid, inescapable rhythms;
But I know, too,
That the blackbird is involved
In what I know.

THE TWO CULTURES

The author and scientist C.P. Snow famously described the modern rift between the humanities and sciences as representing “two cultures”, each having its own distinct vocabulary and methodology, and frequently hostile to the other. (Snow 1964) A similar, if less dramatic, divide exists between the mathematical approaches used to understand cognition and language on the one hand, and most natural phenomena on the other. Cognitive science, linguistics, artificial intelligence, and formal logic have traditionally relied on the use of atomic symbols and the graph structures, grammars, and discrete calculi that operate on them. Electrical engineering and dynamical systems theory (among

[1] Supported by a grant from the Keck Foundation.

[2] Images reproduced with permission from `http://www.chanceandchoice.com`

many other fields) have made use of metric spaces, continuous vectors, and continuous transforms to describe a broad variety of natural phenomena, from sounds and images to population dynamics and the formation of galaxies.

Not surprisingly, the question of how the brain "does" language has been the focus of attention from both of these two camps. Neuroscientists, biologists, physicists, and engineers tend to see the brain as a massively connected signal-processing device operating on continuous-valued quantities (electrical, chemical, and acoustic) in a probabilistic manner. Linguists and cognitive scientists, on the other hand, have typically rejected this level of description as hopelessly inadequate or "low-level" for the phenomena that interest them, most notably the systematic composition of meaning that is the hallmark of human (versus animal) language (Chomsky 1956).

THE CONNECTIONIST ALTERNATIVE

Research in the field connectionist ("neural") networks – the hallmark brain-inspired cognitive systems approach – has made significant inroads into providing a unified computational framework for addressing this issue. Strong criticism of feed-forward network models in the late 1980's (Fodor and Pylyshyn 1988), (Pinker and Prince 1988) has helped fuel the drive for connectionist models that deal adequately with the issues of systematicity (roughly, grammar) and compositionality (roughly, structure) in a principled way. The Simple Recurrent Net (SRN) of Elman has shown impressive abilities in both inducing semantic structure (Elman 1990) and predicting long-distance grammatical dependencies (Elman 1991). For the task of representing explicit compositional structure, the Holographic Reduced Representation (HRR) model of Plate (Plate 2003) has yielded significant insights.

The present work is likewise concerned with addressing the issues of linguistic systematicity and compositionality within a connectionist framework. Unlike the approaches taken by Elman and by Plate, however, our approach is built on a a view of language as a fundamentally self-similar object. [3] To flesh out this approach, we will therefore need at least the three following components: (1) a mathematical model of self-similarity (2) a way of representing primitive linguistic units (words) as primitives of the model (3) a way of representing the composition of these units in terms of the operations provided by the model. Beginning in the next section, we treat each of these requirements in turn.

ITERATED FUNCTION SYSTEMS

Iterated Function Systems provide perhaps the simplest mathematical apparatus for describing self-similar objects. Figure 2 shows a rather lifelike image of a fern plant, where the overall shape of the plant is replicated at various scales (stem, branch, leaf); *i.e.*, the image shows a high degree of self-similarity. This image was generated by the Iterated Function System (IFS) method: starting with an arbitrary set of points in two-dimensional space, the following set of equations was applied iteratively to its own output, until the image no longer changed:

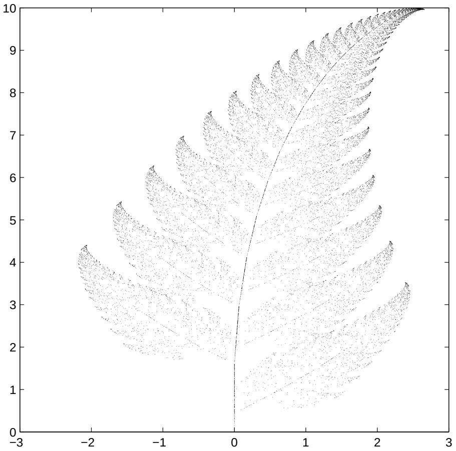

Figure 2: A fern generated by an Iterated Function System. Transforms were applied with the following probabilities: $p(T_1) = 0.01;\ p(T_2) = 0.85;\ p(T_3) = 0.07;\ p(T_4) = 0.07$

$$T_1\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0.16 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$T_2\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} 0.85 & 0.04 \\ -0.04 & 0.85 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 1.6 \end{bmatrix}$$

$$T_3\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} 0.2 & -0.26 \\ 0.23 & 0.22 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 1.6 \end{bmatrix}$$

$$T_4\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} -0.15 & 0.28 \\ 0.26 & 0.24 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0.44 \end{bmatrix}$$

More generally, a given transform can be represented using the formula

$$T_i\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} w_{ixx} & w_{ixy} \\ w_{iyx} & w_{iyy} \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} w_{ix} \\ w_{iy} \end{bmatrix}$$

where the w are weights or coefficients representing scaling, rotation, and translation of the input.

From a practical standpoint, an IFS represents a powerful means of compressing or encoding the digital representation of an image: instead of representing the entire set of points comprising the image – the so-called *attractor* of the IFS – we can store and transmit only the small number of IFS coefficients contained in the four IFS equations, or *transforms*. The actual image can then be reconstructed using the iterative method described above. In principle, this method can be used to encode images of arbitrary dimension – a fact that we will find

[3] We do not mean to imply that SRNs or HRRs are inadequate for dealing with self-similar language structures; rather, we wish to highlight the distinguishing feature of our approach.

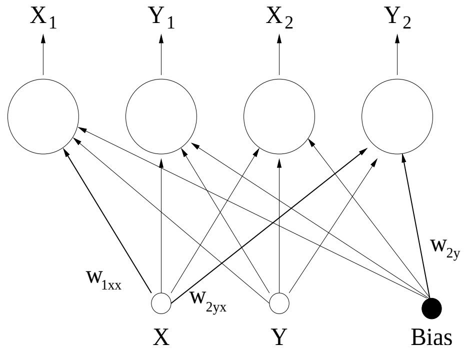

Figure 3: Nonlinear IFS represented as a one-layer neural network. Each output is indexed by the transform applied to obtain it. Sample weights are labeled and highlighted for illustration.

useful later on. It is only our own inability to visualize higher-dimensional objects that has restricted most IFS applications to three or fewer dimensions.

A typical IFS like the one above has transforms that are linear in their input, and so must also be contractive (*i.e.*, a given transform must bring two arbitrary points closer together instead of farther apart); see (Barnsley 1993). By applying a non-linear 'squashing" function to the output of the transforms, however, we can encode images using non-contractive transforms as well. The galaxy-like image in figure 4 was obtained by applying the standard neural-net logistic sigmoid function $f(z) = 1/(1 + e^{-z})$ to the output of the following transforms:

$$T_1 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} -1.2720 & -1.5690 \\ -4.9140 & -1.5150 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2.9640 \\ 4.2720 \end{bmatrix}$$

$$T_2 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} -4.3990 & -2.0180 \\ 5.8130 & -0.7910 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 3.0190 \\ -3.3720 \end{bmatrix}$$

As illustrated in Figure 3, this formulation of an IFS corresponds to a one-layer feed-forward neural network with two input and four output units.

IFS FOR STRUCTURE ENCODING

Iterated Function Systems are generally considered interesting because of their ability to encode fractal images as attractors. There is, however, another interesting property of such systems that can be exploited to encode non-visual information. Each point in the vector space of the IFS is either a member of ("on") the set of attractor points, or "goes to" the attractor via the application of a sequence of one or more transforms. This fact allows us to map from points to fixed-arity tree structures, using the following inductive defintion: (1) each point on the attractor corresponds to a tree of depth zero; (2) each point not on the attractor corresponds to a tree of arity n and depth d, where n is the total number of IFS transforms in the system, and d is the length of the longest sequence of transforms required to take the point to the attractor. This process is illustrated in Figure 5, using the standard parenthesis notation to represent trees.

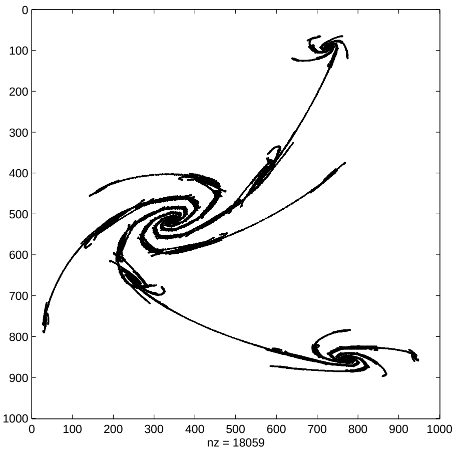

Figure 4: Attractor of a nonlinear IFS. The two transforms were applied with equal probabilities.

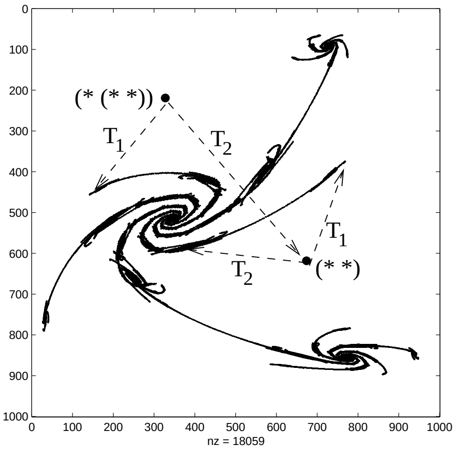

Figure 5: Hypothetical encoding of a depth-two binary tree by a nonlinear IFS. Dashed lines represent IFS transforms. Asterisk represents an arbitrary tree of depth zero.

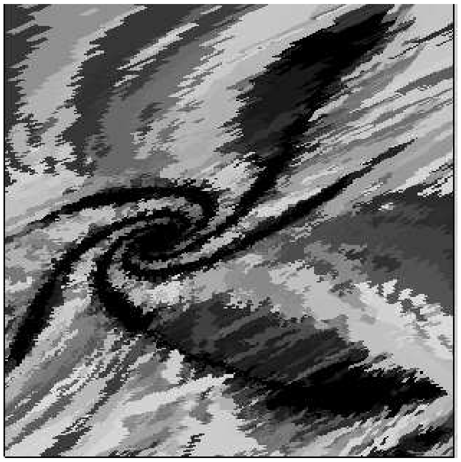

Figure 6: Trees encoded by the Galaxy IFS. Each grayscale level represents a unique tree.

As with images, this fractal tree representation supports the storage of an extremely rich variety of structures in a relatively small code. Figure 6 shows a mapping between points in the vector space of the galaxy IFS and the trees encoded at those points. The figure suggests that this map, like the attractor itself, may be fractal.

FERN LANGUAGES

Structure by itself is of some interest; however, in order to provide a foundation for cognitive domains like language, we must have some way of fleshing out the structure with meaningful content – *i.e.*, we must find some way of satisfying the second of our three requirements above. At minimum, we need a way of associating a depth-zero tree with a symbol, in order to represent, *e.g.*, the difference among the trees $(a\ a)$, $(a\ b)$, $(b\ a)$, and $(b\ b)$, where a and b are arbitrary symbols of the sort used in formal language theory (Hopcroft and Ullman 1979). In terms of the fractal tree encoding described here, we need some way of labeling the attractor points with symbols. We have essentially two choices: either use the IFS itself to label the points, or resort to some external, independently motivated, mechanism. The latter approach is described later in the paper.

Using the former approach, we have shown the ability of networks like the one in Figure 3 to encode exactly the members of the (non-regular) context free language $a^n b^n$(Melnik, Levy, and Pollack 2000). This result agrees with the much more general result obtained in (Tabor 2000), showing the ability of neural networks to act as recognizers for context-free and other formal languages, based on fractally organized computation. We would like to suggest, *in*formally, that such formal languages bear a similar relation to real (natural) languages as fractal ferns and other stereotypical images bear to the general class of images in the real world. Both sorts of mechanism provide a means for capturing important formal properties of objects of interest, but require significant additional mechanisms to "scale up" to real-world applications.[4]

THE COLLAGE THEOREM

To show how fractal image compression can be generalized beyond ferns and the like,, Barnsley (1993) exploits an interesting property of images: although an entire image may not be self-similar, it is likely to contain regions that are similar to each other. For example, although a face does not contain an infinite set of smaller faces, one of the ears may be well approximated by a reflection and translation of the other. Exploiting this property can yield dramatic compression ratios for a variety of real-world images, but it requires the use of many transforms. Each transform is associated with a particular region of the image and mapping that region to another, smaller region. Barnsley's *Collage Theorem* proves that the "correct" set of transforms to yield an attractor for a desired image is one that when applied to the image returns the image itself. Determining the optimal size and shape of the regions (and hence the transform coefficients) is however more of a heuristic process, and several methods are described in the literature (Fisher 1996).

SEMANTIC VECTOR MODELS

Can this approach to compressing real images applied to modeling the structure of real languages? This question brings us back to requirement (2) above; i.e., we need a way of representing primitive linguistic units (words) as primitives of the model. In image compression, the primitives are given by the image itself, as pixels with grayscale or RGB values. For language modeling, we will need some way of associating a vector or set of vectors (point or region) with a particular word or symbol.

Current work in cognitive science provides a variety of vector-based models of word meaning. These models share an inspiration in the insight of Firth (1957) that "you shall know a word by the company it keeps". That is, they build a vector representation for a word based on its pattern of co-occurrence with other words. The dimensionality of the vectors varies from model to model. The Latent Semantic Analysis model (Landauer and Dumais 1997) represents each word using a 300-dimensional vector built using the Singular Value Decomposition of the word co-occurrence matrix. Elman's Simple Recurrent Network (Elman 1990) uses 150-dimensional hidden-layer activation vectors generated by training the network on a next-word prediction task. At the other end of the spectrum, Farkas and Li (2002) build two-dimensional maps of word meaning by running higher-dimensional word co-occurrence vectors through a Kohonen-style Self-Organizing Map (Ritter and Kohonen 1989).

A SIMPLE EXAMPLE

A simple example should suffice to illustrate how we can use co-occurrence vectors as the primitives for IFS tree encodings. We trained a Simple Recurrent Net with two hidden

[4] In evaluating the success of a novel approach to language, such as fractal organization, we should keep in mind that no one has come close to writing a formal grammar for even a (non-trivial) subset of a language like English. At best, we can strive for novel approaches that account for many of the same phenomena as traditional models, while providing additional insights or biological plausibility.

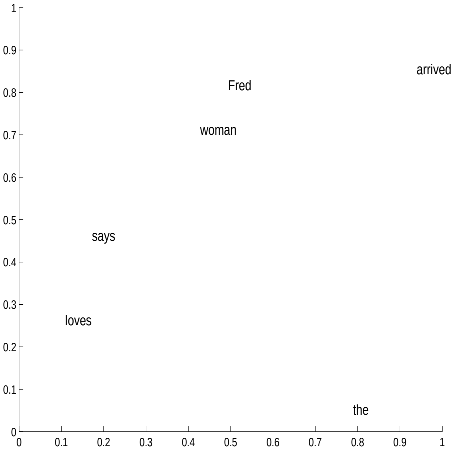

Figure 7: Positions of hidden-layer vectors obtained by a word-prediction tasks.

units on a word prediction task using the following six sentences:

Fred says the woman arrived.
The woman says Fred arrived.
Fred loves the woman.
The woman loves Fred.
The woman arrived.
Fred arrived.

Each of the seven unique symbols (words plus period) was encoded as a one-in-N vector (one bit on, six bits off). Activations were not reset between sentences. Training was performed in on-line mode for 100 epochs with a learning rate of 0.1. Subsequent presentation of the six words on the input layer yielded the hidden-layer activations plotted in Figure 7. The distribution of the vectors shows some of the sort of semantic patterning that Elman found, with nouns (*Fred*, *woman*) close together in one part of the space, and transitive verbs (*loves*, *says*) in another.

Having thus obtained the vector corresponding to each symbol in our training set, we can compose the symbols into trees in a bottom-up manner. Each vector that we wish to make represent a tree is assigned a set of transforms, each of which maps that vector to a vector representing a branch of the tree. Just as determining the location, size and shape of the regions in fractal image compression is a heuristic process involving somewhat arbitrary choices, the choice of which vector should represent a tree is arbitrary.(At this stage of our work we have no intuition about how to choose the vectors, so we simply pick them randomly from the vector space.) Obtaining the transforms for each tree vector is then a simple matter of solving a set of independent linear equations. Figure 8 illustrates this process, using the semantic vectors from the previous example with the tree (*Fred* (*loves* (*the woman*))).

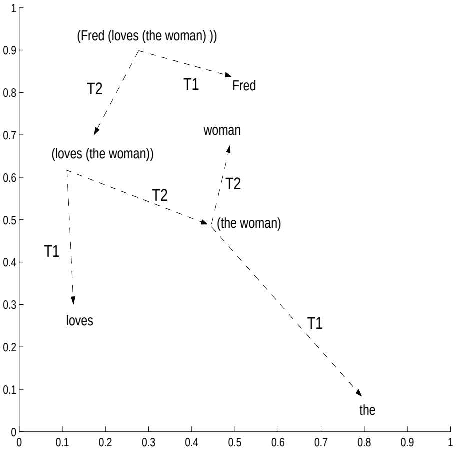

Figure 8: Hypothetical IFS encoding of the tree $(Fred\ (loves\ (the\ woman)))$.

GRAMMAR AS NEURAL COMPRESSION

We have now satisfied the goals set out at the beginning of this paper: we have (1) a mathematical model of self-similarity (2) a way of representing primitive linguistic units (words) as primitives of the model (3) a way of representing the composition of these units in terms of the operations provided by the model. Nevertheless, our solution as described in the previous section is unsatisfying in two important ways: first, it is no longer obvious how the model relates to a brain-inspired cognitive architecture like a neural-network. Second, by assigning a unique set of transforms to each tree vector, we have failed to capture the sort of regularities and generalizations about languages that are expressed by a more traditional model like a context-free grammar. In an important sense, our model is merely a collection of arbitrary transform coefficients.

An important result from neural network theory suggests a principled way out of this situation. It has been known for some time that neural networks with hidden layers can learn arbitrary real-valued vector mappings (White 1990). This fact means that adding one or more hidden layers to the network shown in Figure 3 would allow the network to learn all of the transforms mapping tree vectors to the vectors representing their sub-trees. A hypothetical example of such a network is shown in Figure 9.

Of course, if the number of network weights (*i.e.*, hidden units) required to do this exceeded the number of linear transforms from the non-network version of the model described in the previous section, the network model would not represent any sort of compression, or provide any additional insight. The fact that we were able to obtain network weights to compress a simple formal context-free language (Melnik, Levy, and Pollack 2000) shows that such compression is possible for a highly constrained set of trees over a very small vocabulary. It remains to be seen what sort of generalization can be represented by a network like the one in Figure 9 using a richer variety of trees over many more words.

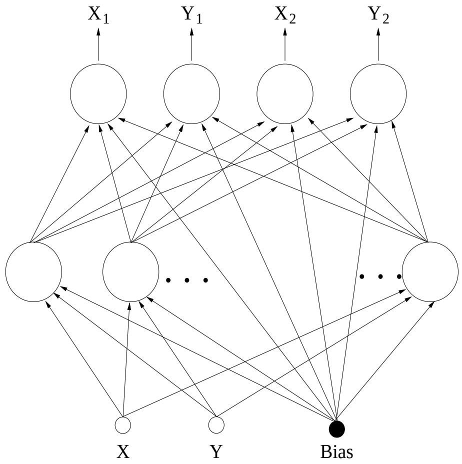

Figure 9: Nonlinear IFS represented as a two-layer neural network.

CONCLUSIONS, IMPLICATIONS, AND FUTURE WORK

Describing the linguistic composition of meaning with fractals instead of grammars allows us to approach a number of important questions in an entirely new way. For example, it is generally agreed that the linguistic data to which children are exposed is of insufficient quality to enable them to induce general structural patterns without some pre-existing mechanism for acquiring language (Chomsky 1965). The traditional approach has been to view this mechanism as a sort of "Universal Grammar" (more accurately, grammar schema) constraining the sorts of languages that human beings can acquire.

Under the approach described in this paper – where the lexicon consists of co-occurrence vectors and the "grammar" is encoded as a set of IFS neural network weights – this "poverty of the stimulus" phenomenon can be viewed as follows: Essentially, the problem is to find a set of tree vectors and network weights such that the frontiers of the trees generated by the IFS match the sentences (strings) to which the learner is exposed.[5] We would like to offer, very tentatively, that the universal mechanism by which such a process might be constrained could be something like Barnsley's Collage Theorem. That is, the "correct" set of tree-vectors and weights could be those that produce an IFS whose attractor covers the set of lexical vectors, so that the IFS effectively maps the lexicon onto itself. The notion that the child explores a set of "candidate grammar" hypotheses while learning language could then be seen as an exploration of the (real-valued) space of network weights. Again, this view is supported by the work of Tabor (2000), who describes how fractal encoding of grammars allows accepting machines for those grammars to be located in a spatial relationship to one another.

An equally compelling issue raised by our approach is how to model parsing, and its relationship to the grammatical "knowledge" encoded in the IFS network.[6] Under a model in which tree structures are encoded as real-valued vectors, parsing becomes the problem of mapping from an input string to a vector encoding the parse tree for that string. Such a process would presumably involve the massively parallel integration of a wide variety of semantic, pragmatic, intonational, and other sorts of information by another neural network (Pollack 1987), leading to the question of how to "compile" the IFS network weights into the weights for the parsing network. An intriguing approach would be to use yet another network to perform (and perhaps learn) this mapping. The ability of a neural network to act as a "parser generator" (Aho, Sethi, and Ullman 1987) of this sort could provide new insights into issues raised by Steedman (2000) about the way in which on-line language processing incorporates knowledge about linguistic structure. As with the learning issue, the contribution would come not from an elimination of explicit structure from the model (Marcus 1998). Rather, it would come from modeling the constraints on this structure in a completely novel way: to paraphrase (Horgan and Tienson 1989), preserving the representations while eliminating the rules.

REFERENCES

Aho, A., R Sethi, and J Ullman (1987). *Compilers: Principles, Techniques and Tools*. Addison-Wesley.

Barnsley, M. F. (1993). *Fractals everywhere*. New York: Academic Press.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on information theory 2*, 113–124.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. Cambridge, Mass.: MIT Press.

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

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

Farkas, I. and P Li (2002). Modeling the development of lexicon with a growing self-organizing map. In *Proceedings of the Fifth International Conference on Computational Intelligence*.

Firth, J. (1957). A synopsis of linguistic theory, 1930-1955. In *Studies in Linguistic Analysis*. Basil Blackwell. reprinted in F. Palmer, ed. *Selected Papers of J. R. Firth.* London: Longman.

Fisher, Y. (Ed.) (1996). *Fractal Image Compression: Theory and Application*. Springer Verlag.

Fodor, J. and Z Pylyshyn (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition 28*, 3–71.

Hopcroft, J. and J Ullman (1979). *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley.

Horgan, T. and J Tienson (1989). Representations without rules. *Philosophical Topics XVII*(1), 147–175.

[5] As Elman (1990) notes, obtaining the co-occurrence vectors requires nothing more complicated than predicting the next word from the current one – a task for which the necessary data are available.

[6] I thank Whit Tabor for forcing me to think about this issue.

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

Mandelbrot, B. (1988). *The Fractal Geometry of Nature*. W.H. Freeman and Company.

Marcus, G. (1998). Rethinking eliminative connectionism. *Cognitive Psychology 37*, 243–282.

Melnik, O., S Levy, and J Pollack (2000). RAAM for an infinite context-free language. In *Proceedings of the International Joint Conference on Neural Networks*, Como, Italy. IEEE Press.

Pinker, S. and A Prince (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition 28*, 73–193.

Plate, T. A. (2003). *Holographic Reduced Representation: Distributed Representation for Cognitive Science*. CSLI Publications.

Pollack, J. (1987). *On connectionist models of natural language processing*. Ph. D. thesis, University of Illinois.

Ritter, H. and T Kohonen (1989). Self-organizing semantic maps. *Biological Cybernetics 61*, 241–254.

Snow, C. P. (1964). *The Two Cultures*. Cambridge University Press.

Steedman, M. (2000). *The Syntactic Process*. Cambridge, Mass.: MIT Press.

Stevens, W., F Kermode, and J Richardson (1997). Wallace stevens : Collected poetry and prose.

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.

White, H. (1990). Connectionist nonparametric regression: multilayer feedforward networks can learn arbitrary mappings. *Neural Networks 3*, 535–550.