Evolving Distributed Representations for Language with Self-Organizing Maps

Simon D. Levy[1] and Simon Kirby[2]

[1] Computer Science Department, Washington and Lee University, Lexington VA 24450, USA, levys@wlu.edu
[2] Language Evolution and Computation Research Unit, School of Philosophy, Psychology and Language Sciences, University of Edinburgh, 40, George Square, Edinburgh, UK

Abstract. We present a neural-competitive learning model of language evolution in which several symbol sequences compete to signify a given propositional meaning. Both symbol sequences and propositional meanings are represented by high-dimensional vectors of real numbers. A neural network learns to map between the distributed representations of the symbol sequences and the distributed representations of the propositions. Unlike previous neural network models of language evolution, our model uses a Kohonen Self-Organizing Map with unsupervised learning, thereby avoiding the computational slowdown and biological implausibility of back-propagation networks and the lack of scalability associated with Hebbian-learning networks. After several evolutionary generations, the network develops systematically regular mappings between meanings and sequences, of the sort traditionally associated with symbolic grammars. Because of the potential of neural-like representations for addressing the symbol-grounding problem, this sort of model holds a good deal of promise as a new explanatory mechanism for both language evolution and acquisition.

1 Introduction

Neural networks hold a great deal of appeal as models of language evolution. As an alternative to traditional "symbol-crunching" systems like grammars, neural nets offer greater biological plausibility – especially with regard to the processing of temporal sequences, limits on structural complexity of meanings, and other "performance" phenomena of real human language. Harnad [1], among others, has argued for the use of neural network models as a solution to the symbol grounding problem, as posed by Searle's famous Chinese Room argument. [2]

A few researchers have successfully used neural networks in modeling language evolution. Typically this work has focused on the emergence of mappings between small, simple meanings and sequences, showing how systematic regularities can emerge in these mappings without using an explicit grammar. A common approach is to embed a neural network in each member of a population of agents, who participate in a communication game over some number of iterations. Batali [3], showed how the back-propagation algorithm [4] can be used to train a population of recurrent neural networks to map from input sequences (abc, cda) to simple propositional meanings (you hungry, me scared). Eventually the agents developed communication systems with structural regularities reminiscent of those in human languages, with a given symbol or sequence of symbols being used to represent the same concept in different contexts.

In a more recent paper, Smith [5] shows how a variant of Hebbian (simple associative) learning can be used to evolve mappings between simple meanings and signals. Each meaning and signal is an unstructured bit vector orthogonal to all the others. By exploring the set of possible learning rules relating signal bit values to meaning bit values, Smith shows how the "innate endowment" and learning biases of communicative agents can result in optimal communication, through a purely cultural (non-genetic) process.

Both of these projects show how insight into language evolution can be gained from even a simple network model. With both projects, however, it is not clear whether or how these results can be extended to more complicated language structures. From a representational perspective, it is not clear how to extend simple binary coding schemes to more complex meanings – especially, how such schemes could represent hierarchical, recursive structures of the sort that appear to underly language and thought. [6]. From an algorithmic perspective, both back-propagation and Hebbian learning pose problems. In addition to being criticized as biologically implausible [7], back-propagation is a computationally intensive, iterative algorithm whose ability to scale up to larger languages is questionable. As for Hebbian learning, the limitations created by the requirement of mutually orthogonal vectors [8] make it unlikely that these sorts of networks would scale up to more realistic, structured representations of meanings and signals.[3]

In the remainder of this paper, we describe a model using a neurally plausible representation of meanings and sequences, and a neural network algorithm for mapping between them, that has the potential to overcome these limitations. We conclude by with some experimental results that validate the ability of this model to learn rule-like mappings, without recourse to grammar.

2 Distributed Representations

In contrast to the the "atomic" or "localist" representations employed in traditional cognitive science, a *distributed representation* is one in which "each entity is represented by a pattern of activity distributed over many computing elements, and each computing element is involved in representing many different entities". [10] Most commonly, the pattern of activity is represented by a vector of real values in some fixed interval, typically $[0, 1]$ or $[-1, 1]$. Proponents of

[3] Subsequent work by Smith *et al.* [9] uses Hebbian associative networks to map structured meanings to structured signals; however, the representation scheme used in that work makes the size of the networks grow explosively as more structure is added, making it impractical for more than very simple structures.

this sort of representations have cited several advantages over traditional symbolic representation. These include robustness to noise ("graceful degradation") and content-addressability (the ability to retrieve items by some feature of their content, rather than an arbitrary address), which are properties of human intelligence seen as lacking in traditional symbolic models [8].

Distributed representations of meaning have appeared in a variety of contexts in contemporary AI and cognitive science. Most commonly they are used to model the meanings of individual words. In a widely cited paper, Elman [11] demonstrated the ability of a simple recurrent neural network to form richly structured distributed representations of word meaning, based on the task of predicting the next word in a sentence. More recently, the method of Latent Semantic Analysis [12] has used distributed representations successfully in a wide variety of practical AI tasks. On a more theoretical level, Gärdenfors [13] has elaborated a framework in which conceptual meanings are analyzed as regions in a vector space. A very useful feature of all such models is that the vector representations of similar structures end up close together in the vector space, as determined by a common metric like Euclidean distance, dot product, or cosine.

Although these sorts of distributed representations can be seen as encoding structure, it is structure of a categorical, rather than propositional or sentential, nature. As pointed out by Steedman [14], such structure corresponds more to part-of-speech information than to the propositional structures used in AI, logic, and linguistics. For example, given distributed representations of the concepts man, tiger, and chases, simply adding or multiplying the representations together gives no way to extract the difference between the propositions chases(man,tiger) and chases(tiger,man); but these propositions contrast in the assignment of the agent and patient roles to each of the two arguments.

Partly in response to such criticisms, several researchers have developed distributed representations of structured meaning. These include the Holographic Reduced Representations (HRR) of Plate [15], the Binary Spatter Codes of Kanerva [16], the Context-dependent Thinning Networks of Rachkovskij [17], and the Multiplicative Binding Networks of Gayler [18]. All these architectures use vectors of real (or binary) values with high dimensionality (typically 1000 or more dimensions), a binding operation to join vectors representing roles (agent, patient) with those representing fillers (man, tiger), and a combinatory operation to build meaningful structures out of the bound elements.[4] Crucially, these operations do not increase the size of the representations, which was a problem in earlier distributed representation binding schemes. [20]

In Plate's HRR framework, used in our experiments reported below, the binding operation is circular convolution: given vectors $\tilde{c}$ and $\tilde{x}$ of dimension n,

[4] Pollack's Recursive Auto-Associative Memory [19] is a close cousin of such representations, using relatively low-dimensional vectors for fillers, and matrices for roles.

their circular convolution “trace vector” $\tilde{t} = \tilde{c} \circledast \tilde{x}$ is defined as

$$t_j = \sum_{k=0}^{n-1} c_k x_{j-k} \tag{1}$$

for $j = 0$ to $n-1$, subscripts modulo-n. A noisy version $\tilde{y}$ of $\tilde{x}$ can be recovered by circular correlation: $\tilde{y} = \tilde{c} \mathbin{\#\!\!\!\bigcirc} \tilde{t}$,defined as

$$y_j = \sum_{k=0}^{n-1} c_k t_{k+j} \tag{2}$$

for $j = 0$ to $n-1$, subscripts modulo-n. The distributed vector representation of a proposition like chases(tiger,man) can then be computed as

$$R(\mathsf{chases(tiger,\ man)}) = R(\mathsf{chases}) \;+\; R(\mathsf{tiger}) \circledast R(\mathsf{agent}) \;+\; R(\mathsf{man}) \circledast R(\mathsf{patient}) \tag{3}$$

where $R(\mathit{symbol})$ is the distributed representation of *symbol*. The representation in (3) encodes both the fact that the proposition is about chasing (first term) and the fact that it is the tiger doing the chasing and the man being chased (last two terms). To query, *e.g.*, who did the chasing in this representation, we correlate the sum in (3) with $R(\mathsf{agent})$, and compare the noisy result with each of the original symbol vectors, to see which is closest (the so-called “cleanup” operation). As long as the original vectors are chosen randomly (zero mean, variance $1/n$), and given a sufficiently large n, this scheme can be used to encode arbitrarily complex structures like knows(man, believes(woman, chases(tiger, man))).

For language evolution research, we also need a way of representing symbol sequences. Plate [15] describes several ways of representing sequences with HRR. In the work described below, we use the method of positional cues, in which a separate set of vectors $\tilde{p}_i$ encodes the position of each element in the sequence by means of the convolution operation:

$$R(\langle \mathsf{a,b,c} \rangle) = \tilde{p}_1 \circledast R(\mathsf{a}) \;+\; \tilde{p}_2 \circledast R(\mathsf{b}) \;+\; \tilde{p}_3 \circledast R(\mathsf{c}) \tag{4}$$

A noisy version of the i^{th} sequence element can be recovered from the distributed representation of the sequence by circular correlation with $\tilde{p}_i$. For example:

$$R(\mathsf{a}) \cong \tilde{p}_1 \mathbin{\#\!\!\!\bigcirc} R(\langle \mathsf{a,b,c} \rangle) \tag{5}$$

As with the distributed representations of concepts discussed in the previous section, HRR and related coding schemes have the feature that the vector representations of similar structures (chases(tiger,man), chases(lion,man)) end up close together in the vector space. This fact is illustrated in Table 1, for a set of simple propositions containing a predicate (arbitrarily denoted by p, q, and r) and one argument (arbitrarily denoted by x, y, and z) . The same property holds for the vector representations of similar sequences.

With efficient distributed representations of arbitrarily complex meanings and signals, we arrive at the question of how to evolve mappings between the

Table 1. Cosines between 1000-dimensional HRR's of simple propositions

	p(x)	p(y)	p(z)	q(x)	q(y)	q(z)	r(x)	r(y)	r(z)
p(x)	1.00								
p(y)	0.32	1.00							
p(z)	0.31	0.28	1.00						
q(x)	0.71	0.04	0.03	1.00					
q(y)	0.01	0.69	-0.01	0.32	1.00				
q(z)	0.01	0.00	0.70	0.31	0.31	1.00			
r(x)	0.72	0.06	0.04	0.70	0.03	0.02	1.00		
r(y)	0.04	0.71	0.01	0.04	0.69	0.01	0.35	1.00	
r(z)	0.04	0.02	0.71	0.03	0.01	0.70	0.33	0.32	1.00

two. An obvious approach would be to train a three-layer backpropagation network to perform the mapping. This approach would however suffer from the problems described in relation to backprop network earlier: training times can grow arbitrarily long, and the algorithm itself lacks biological plausibility. The following section reviews the Kohonen Self-Organizing Map, the neural-net architecture that we ended up choosing for this task.

3 Kohonen's Self-Organizing Map

The Self-Organizing Map (SOM) of Kohonen [21] is an unsupervised neural network learning method that can be used to reveal patterns of organization in a data set. The data set X consists of vectors of a fixed dimensionality. The network is typically organized into a two-dimensional grid $U_{i,j}, 1 \leq i \leq m, 1 \leq j \leq n$ of nodes, each of which is associated with an initially random weight vector $\tilde{w}^{i,j}$of the same dimensionality as the members of X. On each learning iteration, a vector $\tilde{x}$ is randomly chosen from the data set, and the node whose weight vector is closest to this vector is considered the "winner" for that iteration. The winning node's weight vector is updated to move it closer to the vector picked from the data set, as are the weight vectors of the winner's grid neighbors. By decreasing the neighborhood size with increasing iterations, the weight vectors eventually settle into a reasonable representation of the data set.

Figure 1 shows a simple example of SOM learning. Here, the data set is two-dimensional, so each grid point is associated with a two-dimensional weight vector. The data set consists of points sampled uniformly from a ring shape. Each grid node is plotted at the point corresponding to its weight vector, and is connected to its north, south, east, and west neighbors by a line segment. The figure shows that no matter how close together or far apart the weight vectors are initially, they end up distributing themselves (and their associated nodes) uniformly within the space enclosing the ring shape.

A common application of SOM is dimensionality reduction for data visualization in two dimensions. There is, however, no restriction on the dimensionality of the nodes U. In fact, the grid of nodes is itself a special case (discrete, two-dimensional) of a continuous metric space, and the algorithm will work with any

U for which a neighborhood (distance) metric is defined. The $U_{i,j}$ are replaced with vectors $\tilde{u}^i, 1 \leq i \leq n$, and the index k of the winner $\tilde{u}^k$ is defined as

$$k = \arg\min_i |\tilde{w}^i - \tilde{x}| \tag{6}$$

where $\tilde{w}^i$ is the weight vector associated with $\tilde{u}^i$, and $|\tilde{x} - \tilde{y}|$ is the distance between $\tilde{x}$ and $\tilde{y}$. Instead of updating the winner and the nodes in its neighborhood, *all* nodes in the network are updated, with the size of the update determined by distance from the winner:

$$\tilde{w}^i_{t+1} \leftarrow \tilde{w}^i_t + \mu_t f(i,k,t)(\tilde{x} - \tilde{w}^i_t) \tag{7}$$

where μ_t is a learning rate parameter, f is the neighborhood function

$$f(i,k,t) = e^{-|\tilde{u}^i - \tilde{u}^k|^2 / 2\sigma_t^2} \tag{8}$$

and σ_t is a neighborhood parameter. Both parameters decrease with time, allowing the weights to settle into an approximate representation of the data set X. In short, the SOM forms a regular *topographic map* [22] from one vector space to

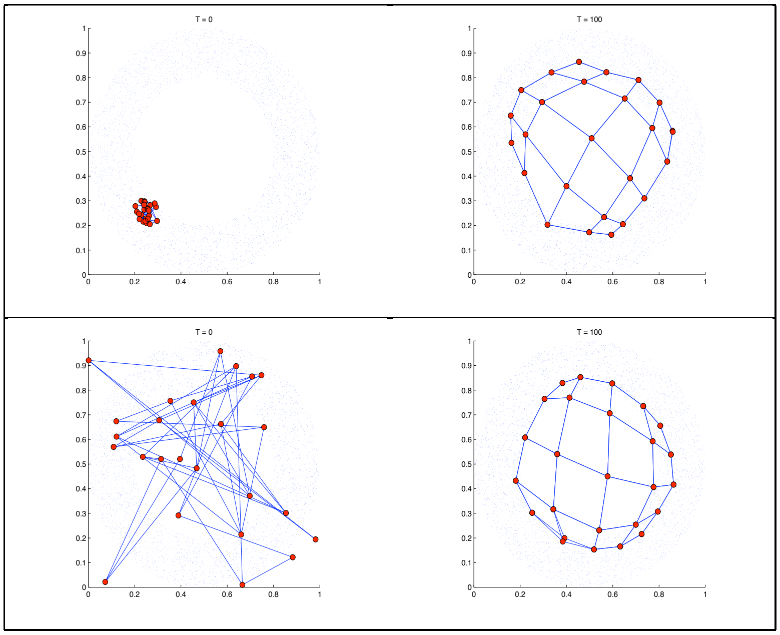

Fig. 1. Two-dimensional SOM learning a ring shape. Final (T=100) configurations are similar regardless of whether initial (T=0) weights are clustered close together (top) or far apart (bottom).

another. Recent work in language evolution [23] has argued for the importance of such maps as a key to understanding the ways in which languages develop and change.

With this broader understanding of SOM, it becomes possible to develop efficient mappings between high-dimensional distributed representations of symbol sequences and high-dimensional distributed representations of meanings. Each meaning vector $\tilde{u}$ can be associated with a weight vector $\tilde{w}$. The sequence vector $\tilde{x}$ expressing $\tilde{u}$ is then chosen as the member of the vector space X of possible sequences that is closest to $\tilde{w}$. In the next section we describe an algorithm that uses this scheme to evolve systematic, grammar-like mappings.

4 Experiments

4.1 Learning a Simple Mapping

To explore the possibility of using an SOM to evolve systematic mappings between distributed representations of meanings and sequences, we started with a simple model not explicitly involving agents or communication. We used the small predicate/argument propositional meanings from Table 1, and symbol sequences of length two over the alphabet {a, b, c}. Meanings were represented as 1000-dimensional HRR trace vectors, and sequences as 1200-dimensional HRR trace vectors. (These sizes were chosen arbitrarily, to show that the meaning traces and sequence traces need not agree in size.) Each 1000-dimensional meaning trace was associated with an initially random 1200-dimensional weight vector, which was modified according to Equations 6 - 8, with the meaning vectors being the U, the sequence vectors the X, and the weights between them the W. The learning rate μ_t was scaled linearly from 0.5 to 0.1, and the neighborhood value σ_t from 3.0 to 0.1. Our goal was to see what sorts of meaning-sequence mappings emerged.

Table 2 shows the results of eight different experimental runs of 500 SOM learning iterations each. In each row i, the first column shows the i^{th} propositional meaning. The second column shows the "winning" sequence for that meaning at the start of the experimental run; *i.e.*, sequence j whose sequence trace $\tilde{x}^j$ is closest to the weights $\tilde{w}^i$ for that meaning:

$$j = \arg\min_k |\tilde{w}^i - \tilde{x}^k| \qquad (9)$$

The remaining columns show the winning sequences at the end of eight different experimental runs. As Table 2 indicates, the meaning-sequence mappings changed from being highly non-systematic at the beginning of the experiment to maximally systematic at the end, for all but one of the eight runs reported. Each randomly initialized SOM learned to map from a given predicate (p, q, or r) to a single symbol (a, b, or c), and from a given argument (x, y, or z) to a single symbol. For example, the last column in the table shows a "verb-final" mapping in which the symbol corresponding to the predicate comes second, the symbol corresponding to the argument comes first, and the systematic mappings

Table 2. Results of First Experiment

	Typical Initial								
Meaning	Sequences	Final Sequences							
p(x)	ac	bb	ca	cb	cc	ac	cc	ba	ba
p(y)	ac	cb	ba	bb	ac	cc	bb	aa	ca
p(z)	ac	ab	aa	ab	bc	bc	ba	ca	aa
q(x)	ac	bc	cc	cc	ca	ab	cc	bc	bc
q(y)	ac	cc	bc	bc	aa	cb	bc	ac	cc
q(z)	ac	ac	ac	ac	ba	bb	ca	cc	ac
r(x)	ca	ba	cb	ca	cb	aa	ac	bb	bb
r(y)	ac	ca	bb	ba	ab	ca	ab	ab	cb
r(z)	bc	aa	ab	aa	bb	ba	aa	cb	ab

are $(p/a,\ q/c,\ r/b)$ for predicates and $(x/b,\ y/c,\ z/a)$ for arguments. These results show that our approach can produce systematic mappings, for this small learning task at least.

4.2 Opening the Bottleneck

The mappings learned in the previous experiment are, for the most part, *compositional*: a given meaning component (predicate or argument) is always represented by the same symbol, independent of where it appears. No two predicates are represented by the same symbol, nor are any two arguments. This situation led us to ask whether our HRR/SOM learning model has a bias toward compositionality, or whether there is some other influence at work.

To examine this issue, we repeated the first experiment with an alphabet of six symbols instead of three. If our model were biased toward maximal compositionality, we would expect to end up with a one-to-one mapping between each meaning element and each sequence symbol. After trying a number of parameter settings, we were unable to obtain compositional mappings for this experimental setup. An example final, non-compositional sequence is shown in Table 3. A look

Table 3. Lack of Compositionality

	Typical Initial	Typical Final
Meaning	Sequences	Sequences
p(x)	ee	af
p(y)	ee	fd
p(z)	ee	ed
q(x)	ee	ab
q(y)	ff	cc
q(z)	be	dd
r(x)	ee	ba
r(y)	de	cd
r(z)	be	dc

back at Figure 1 suggests a possible explanation for this lack of compositionality. This figure shows that, regardless of the initial weights, SOM learning produces a final weight configuration that is evenly distributed around the space defined by the input data. In the first experiment, the number of sequences was identical to the number of meanings. Hence, the there was no "room" in the input space for the weights to expand, and this even distribution yielded a compositional mapping. In the second experiment, there were four times as many sequences (36) as meanings (nine). By distributing the meanings throughout the space of sequences, the SOM produced a highly non-compositional mapping for this data set.

This result may be seen as analogous to the *bottleneck* principle described by Kirby [24], in which the constraints of cultural transmission favor the emergence of languages describable by a small number of rules. The previous two experiments show how the constraint imposed by using a smaller number of symbols results in a similar outcome, using an entirely different computational substrate.

4.3 Evolving Word-Order Regularities

With this understanding of our HRR/SOM model in mind, we turned our attention to using the model to study specific phenomena. In a third experiment, we used the model to explore the emergence of word-order regularities among the subject, object, and verb in a simple model of sentence production. Based on a data set used by Kirby [24], we constructed simple proposition meanings of the form $predicate(argument1, argument2)$, where $predicate$ ranged over {loves, hates, admires, sees, detests}, and each argument ranged over {john, mary, gavin, bill, fred}. For the sake of clarity in comparing the relative order of subject and object, we avoided reflexives, yielding 100 $(5 \times 5 \times 4)$ propositional meanings. Using the symbol set {l, h, a, s, d, j, m, g, b, f}, we constructed all six permutations of compositional three-symbol "sentences" for each such meaning; for example, the proposition loves(john,mary) yielded the possible sentences {ljm, lmj, jlm, jml, mjl, mlj}. Meanings and sequences were both represented by 2000-dimensional HRR trace vectors. Unlike the previous two experiments, this experiment associated the weight vectors to the *sequences*, rather than the meanings, resulting in a situation in which six possible sequences were competing for each meaning. The winner of each competition was chosen via Equation 6, after which the weights for all 600 sequences (not just the six competitors) were updated via Equation 7. For this experiment the learning rate μ_t decreased linearly from 0.125 down to 0.025.

The results of this experiment were quite consistent: over 500 iterations of the SOM learning algorithm, the astronomically large set of possible mappings quickly converged to one of the six possible word orders relating predicates and arguments to verbs, subjects, and objects (VSO, VOS, SVO, SOV, OSV, OVS). Figure 2 shows a sample experimental run, where the model converged to SVO word order. The figure shows the fraction per 10 iterations of each kind of mapping. Note that the SVO order becomes dominant before 50 iterations have passed, meaning that the model begins to generalize before fewer than half of the possible meanings have been presented to it.

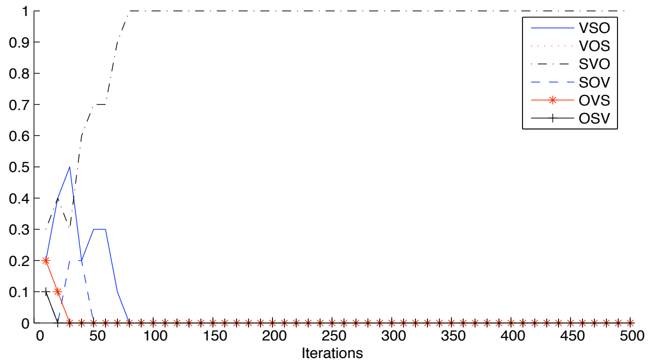

Fig. 2. Sample run for the third experiment, showing fraction of mappings with given word order against iterations, in steps of 10.

5 Discussion

The work described here represents a very preliminary attempt to provide a neurally plausible alternative to traditional grammars as a basis for research in language evolution and development. Our model learns to map between high-dimensional, distributed representations of propositional meanings and symbol sequences, without using a computationally expensive and biologically implausible algorithm like back-propagation.

Our goal is not to supplant existing approaches to explaining the development of language. Indeed, the very nature of our model, in which symbol sequences compete for the propositional meanings that they signify, is very much in the spirit of several modern approaches to these issues. The most recent and obvious of these is the evolutionary approach of Croft [25], in which individual "linguemes" (phonemes, morphemes, words, phrases, collocations) are viewed as competing for usage in a speech community. Alternatively, our model could be viewed in the context of an individual language learner who, presented with a small, finite set of exemplars, must generalize to the patterns of the full language [26]. Figure 2 suggests this sort of capability. What we hope to add to these endeavors is a sense of how the symbolic representations used in all of them might be grounded in a neurally plausible model of representation. By using these sorts of representations throughout – instead of merely at the lowest sensory/motor level – we avoid the grounding problem associated with traditional symbol systems.

As with any new model, however, we have of necessity ignored a number of crucial issues. Most glaring of these is perhaps our treatment of sequences, in which we encode the absolute position of each symbol. A more psychologically realistic model would focus on the relative position of symbols, thereby supporting the kinds of phenomena found in serial-order experiments. [27] Nor have we dealt in any way with recursion, a property generally considered to be part of the minimally adequate characterization of human language. [28] As noted above, a desirable feature of the Holographic Reduced Representations employed here is their ability to encode recursive structures of arbitrary complexity. Another possible direction for this research would therefore involve exploring the kinds of mappings that emerge from the need to communicate recursive propositional meanings with symbol sequences, using the HRR/SOM model.

Finally, the fact that this work does not involve a direct model of transmission of information between agents leads to the question of whether our model can be accommodated in a more traditional agent-based approach. A test of this ability would be to force the association of particular meaning/sequence pairs, and then see whether the model could self-organize the remainder of the unseen meanings – as suggested by the data shown in Figure 2. Preliminary results show that this is indeed the case, supporting the extension of our model to an agent-based setting.

Acknowledgments

The first author would like to thank: Washington and Lee University for support during the sabbatical leave in which the ideas in this paper were developed; Mark Steedman and the University of Edinburgh Informatics Department for sponsoring his visit; Jim Hurford, Kenny Smith, and the Language Evolution and Computation Research Unit at Edinburgh for discussion and criticism about the ideas presented here; and Tony Plate for help with the HRR model.

References

[1] Harnad, S.: Grounding symbols in the analog world with neural nets. Think **2**(1) (1993) 12–78

[2] Searle, J.: Minds, brains, and programs. Behavioral and Brain Sciences **3** (1980)

[3] Batali, J.: Computational simulations of the emergence of grammar. In Hurford, J., Studdert-Kennedy, M., Knight, C., eds.: Approaches to the Evolution of Language: Social and Cognitive Bases. Cambridge University Press, Cambridge (1998)

[4] Rumelhart, D., Hinton, G., Williams, R.: Learning internal representation by error propagation. In Rumelhart, D., McClelland, J., eds.: Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1. MIT Press (1986)

[5] Smith, K.: The cultural evolution of communication in a population of neural networks. Connection Science **14**(1) (2002) 65–84

[6] Fodor, J.: The Language of Thought. Crowell, New York (1975)

[7] Grossberg, S.: Competitive learning: from interactive activation to adaptive resonance. In: Connectionist models and their implications: readings from cognitive science. Ablex Publishing Corp., Norwood, NJ, USA (1988) 243–283

[8] McClelland, J., Rumelhart, D., Hinton, G.: The appeal of parallel distributed processing. In Rumelhart, D., McClelland, J., eds.: Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1. MIT Press (1986)

[9] Smith, K., Brighton, H., Kirby, S.: Complex systems in language evolution: the cultural emergence of compositional structure. Advances in Complex Systems **6**(4) (2003) 537–558

[10] Hinton, G.: Distributed representations. Technical Report CMU-CS-84-157, Computer Science Department, Carnegie Mellon University (1984)

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

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

[13] Gärdenfors, P.: Conceptual Spaces: The Geometry of Thought. MIT Press (2000)

[14] Steedman, M.: Connectionist sentence processing in perspective. Cognitive Science **23**(4) (1999) 615–634

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

[16] Kanerva, P.: The binary spatter code for encoding concepts at many levels. In Marinaro, M., Morasso, P., eds.: ICANN '94: Proceedings of International Conference on Artificial Neural Networks. Volume 1., London, Springer-Verlag (1994) 226–229

[17] Rachkovskij, D.A., Kussul, E.M.: Binding and normalization of binary sparse distributed representations by context-dependent thinning. Neural Computation **13**(2) (2001) 411–452

[18] Gayler, R.: Multiplicative binding, representation operators, and analogy,. In Holyoak, K., Gentner, D., Kokinov, B., eds.: Advances in Analogy Research: Integration of Theory and Data from the Cognitive, Computational, and Neural Sciences. New Bulgarian University, Sofia, Bulgaria (1998) 405

[19] Pollack, J.: Recursive distributed representations. Artifical Intelligence **36** (1990) 77–105

[20] Smolensky, P.: Tensor product variable binding and the representation of symbolic structures in connectionist systems. Artificial Intelligence **46** (1990) 159–216

[21] Kohonen, T.: Self-Organizing Maps. 3 edn. Springer-Verlag, Secaucus, NJ (2001)

[22] VanHulle, M.: Faithful Representations and Topographic Maps. Wiley-Interscience, New York (1990)

[23] Brighton, H., Kirby, S.: Understanding linguistic evolution by visualizing the emergence of topographic mappings. Artificial Life **12**(2) (2006) 229–242

[24] Kirby, S.: Learning, bottlenecks and the evolution of recursive syntax. In Briscoe, T., ed.: Linguistic Evolution through Language Acquisition: Formal and Computational Models. Cambridge University Press (2002)

[25] Croft, W.: Explaining language change: an evolutionary approach. Longman, Harlow, Essex (2000)

[26] Chomsky, N.: Rules and Representations. Basil Blackwell, Oxford (1980)

[27] Lewandowsky, S., Murdock, B.: Memory for serial order. Psychological Review **96**(1) (1989) 25–27

[28] Hauser, M.D., Chomsky, N., Fitch, W.T.: The faculty of language: What is it, who has it, and how did it evolve? Science **298** (2002) 1569–1579