To be published in: *Approaches to the Evolution of Language: Social and Cognitive Bases*, edited by James R Hurford, Michael Studdert-Kennedy and Chris Knight, Cambridge University Press, 1998

Computational Simulations of the Emergence of Grammar

John Batali

Department of Cognitive Science
University of California at San Diego
9600 Gilman Drive
La Jolla, CA 92903-0515
batali@cogsci.ucsd.edu

Abstract

A model of simple agents capable of sending and receiving sequences of characters and associating them with elements of a set of structured meanings is used to explore the emergence of systematic communication. In computational simulations, each member of a population alternates between learning to interpret the sequences sent by other members, and sending sequences that others learn to interpret. Eventually the agents develop highly coordinated communication systems that incorporate structural regularities reminiscent of those in human languages.

1 Introduction

Human language makes it possible to express a vast number of different and complex meanings with sequences composed of a relatively small number of relatively simple elements — and to interpret such sequences as the meanings they express. A traditional sense of the word "grammar," and the one I adopt here, refers to the systematic regularities between meanings and their expression as sequences of sounds or gestures in a language.

In particular, the grammars of human languages incorporate systematic regularities involving the *structures* of the meanings and the sequences that express them. For example the morphological structure of an inflected verb conveys information about the person, number, and gender of one or more of the participants in the event or situation the verb describes.

The structural regularities of a language constitute a resource that a speaker can use to express novel meaning combinations, provided that the elements out of which the new meaning is constructed and the relations constituting that structure are consonant with the language's grammar. The hearer can accurately interpret the utterance as involving those familiar structural constituents and relations, even though that specific combination may have never been used before. Thus the system can be used to express meanings tailored to the specific occasion of their use, and can be extended as the need to express new meanings arises. Learning is relatively easy because the novice has to master only the structural regularities and basic elements of a language, instead of memorizing all of its meaning/expression pairs.

The ability to communicate by exploiting a system of structural regularities therefore represents an invaluable achievement of a species for which coordinated social activity is vital to survival, and for which the accurate exchange of information often provides adaptive benefit. Given this benefit, it is tempting to explain the achievement as the result of natural selection.

However it is important to distinguish between the evolution of language itself — in particular the emergence, modification, and enrichment of the grammatical resources in human languages — and the biological evolution of articulate hominids. Clearly they are related: Lacking appropriate anatomical and neurological endowment, an animal will be unable to produce or perceive complex signals, and without sufficiently powerful cognitive abilities, it cannot entertain meanings worth communicating in the first place. But the adaptive benefits of such traits are not specific to communication, and it is not clear how communication alone could provide sufficient selection pressure for their development.

In this paper I explore the idea that some of the grammatical regularities manifest in human language could emerge as a result of non-genetic cultural processes among a population of animals with the cognitive capacities required for communication, but who do not initially share a coordinated communication system. Whether or not this is what really happened in our species is unknown, but the possibility seems worth investigating, to better

understand both its plausibility and its limitations.

Speculation about the origins and early development of human language must perforce originate in intuitions based on experience with its modern versions. Though these intuitions can be tested methodologically, with the resources of linguistics, psychology, anthropology, neuroscience, literary studies, and other disciplines, we are for the most part limited to the one class of exemplars. There is obviously no way to go back and observe what happened since the Pliocene epoch, and the differences between the regularities exhibited by the grammars of human languages and those of other animal communication systems seem profound.

Mathematical and computational models provide a way to explore alternative accounts of the emergence of systems of communication. If the consequences of a model are consistent with expectations based on intuitions or speculation, they might obtain some small measure of support. But more interestingly (and, as it happens, more often), the consequences of a model may deviate from expectations. In working out the reasons for the differences, one can potentially develop a richer set of intuitions. Models are thus valuable to the degree that they explicitly illustrate the consequences of the set of assumptions they embody. This may be even more important than whether those assumptions are correct.

The computational simulations described in this paper involve populations of simple agents that can produce sequences of tokens to encode structured meanings, and can assign interpretations to sequences of tokens. Initially the agents' communication systems are almost totally uncoordinated: Few, if any, agents send the same sequence for the same meaning, and none of the agents is able to correctly interpret sequences sent by others. During the simulation runs, each agent alternates between learning to interpret sequences sent by the other members of the population, and sending sequences that other agents learn to interpret. Such populations eventually develop systems that support highly accurate communication, even of novel meaning combinations. As the cognitive skills of the simulated agents are possessed by some primate species, and no language-specific innate capacities are required, the simulations might model the emergence of some of the grammatical regularities in human languages.

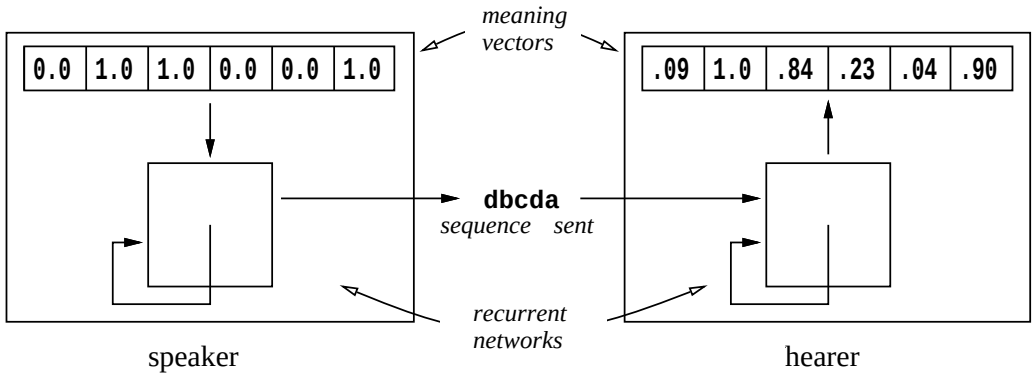

Figure 1: Communicative agents. The values in the speaker's meaning vector are used by the speaker's recurrent network to determine a sequence of characters to send. The hearer's recurrent network processes that sequence to determine the values stored in the hearer's meaning vector.

2 Communicative Agents

The simple model of communication explored in the computational simulations is illustrated in Figure 1. Each agent contains a "meaning vector" that stores ten real numbers between 0.0 and 1.0, and a simple recurrent network that is used to send and receive sequences of characters from the set $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$.

In an episode of communication between a pair of agents, each value in the speaker's meaning vector is first set to either 0.0 or 1.0, depending on which of the set of meanings described in Section 3 is to be conveyed. The values in the speaker's meaning vector are used by the speaker's recurrent network to determine the sequence of characters it sends. This sequence is processed by the hearer's network to determine the hearer's meaning vector.

The accuracy of a communicative episode is assessed by comparing the values on the speaker's meaning vector with those in the hearer's after the sequence sent by the speaker for a given meaning has been processed by the hearer. A value in the hearer's vector will be called "correct" if it is within 0.5 of the value in the corresponding position of the speaker's vector. A more sensitive measure of the accuracy of the hearer's interpretation is obtained by computing the root mean square of the difference between the values in

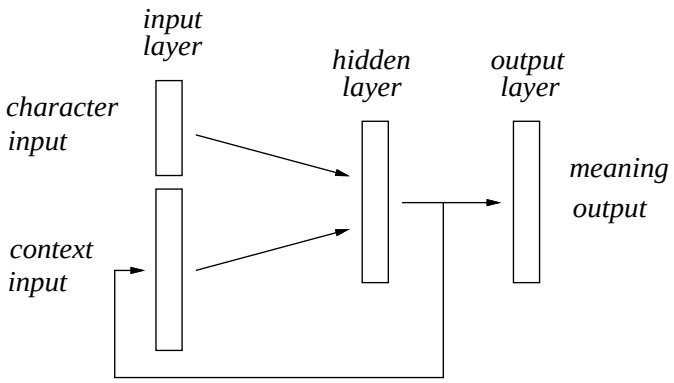

Figure 2: Recurrent neural networks used in the communicative agents. Each layer contains a set of units whose activation values are determined by units in the previous layer and the values of connection weights between the units. Activation values of the character input units in the input layer are set externally. Activation values of the context input units are copied from from the activation values of the units in the hidden layer after each character is processed.

meaning vectors of the speaker and the hearer, after processing the sequence. This value will be referred to as the hearer's "error" for the sequence.

Treating meaning as a pattern of binary values is wildly simplistic of course, but is at least straightforward and explicit, and is consistent with the approach of information theory. A more subtle assumption underlying this model is that the agents are capable of producing and recognizing tokens from some finite set, and of mapping sequences of such tokens to and from meanings. While there is evidence that humans perceive and produce speech sounds as tokens of discrete categorical types, and that this ability is partly innate, it is also at least partly learned (Eimas et al., 1971). Even if innate, the ability to perceive such categories must have developed along with other linguistic abilities, as opposed to being present in fully developed form before the emergence of grammar, as this model assumes.

The architecture of the recurrent network in each agent is is illustrated in Figure 2. The networks have three layers of units, with feed-forward connections between the units of the input layer and those of the hidden layer, and between the hidden layer and the output layer. The logistic activation

h.1 Set all network inputs to zero.

h.2 Process each character:

- **h.2.a** Set the input corresponding to the character to 1.0; set the rest of the inputs to 0.0.
- **h.2.b** Feed activation values forward through the network.
- **h.2.c** Copy the values of the hidden units to the context input units.

h.3 When the sequence is complete, copy the values of the output layer of the network to the agent's meaning vector.

Figure 3: Operation of the recurrent networks shown in Figure 2, when receiving a sequence of characters.

function used for all of the units in the network. (See Haykin, 1994, Chapter 6.) Each network has one input unit for each of the characters, thirty context input units, thirty units in its hidden layer, and ten output units (corresponding to the number of values in the agents' meaning vectors).

When receiving a sequence of characters (i.e., when used inside the "hearer" of Figure 1), the networks are operated as described in Figure 3. After initializing the network, each character in the sequence is processed by activating the input unit corresponding to that character, feeding activation values forward through the network, and then copying the activation values of the hidden layer to the network's context input units. These values can thus encode temporal properties of the sequence that has been processed so far. After the last character of the sequence has been processed, the output of the hearer's network represents the hearer's interpretation of the sequence.

Recurrent neural networks can be trained to associate specific output vectors with specific sequences, or with sequences that satisfy various formal constraints. (Jordan, 1986; Cleeremans et al., 1989 Elman, 1990; Siegelmann, 1993; Batali, 1994.)

To train a network to interpret a sequence as a given meaning vector, the network is operated as shown in Figure 3, except that after step h.2.b,

the backpropagation algorithm (Rumelhart, et al., 1986) is used to modify the weights of the network. The error of each output unit is determined by the difference between the value of the meaning vector at the position corresponding to the unit, and the unit's actual value. A backpropagation learning rate of 0.01 is used.

The network is trained after each character in the sequence is processed, even the first character. Though, in general, it won't be able to correctly interpret sequences so early, this method of training forces the networks to develop representations of temporal properties of the sequences in their hidden layers that will enable the networks to produce the correct output after more characters are processed.

The agents' networks are used to receive sequences according to the algorithm described in Figure 4. As the speaker generates a sequence of characters, its network processes those characters as if it were receiving them. To choose which character to send at each point in the sequence, the speaker determines which of the four characters would bring its own output closest to the meaning being conveyed. That character is then sent, and processed, by the speaker. If, after doing so, all the speaker's output units are correct for the meaning being conveyed (or if the sequence has reached a cutoff length of twenty characters), the speaker stops sending. Otherwise the process is repeated.

Networks are not trained to send sequences for meanings. However being trained to interpret sequences will also modify a network's sending behavior by changing the network's connection weight values.

The mechanism for sending characters was designed with the results of Hurford (1989) in mind. His simulations involve a simpler model of communication, in which unitary signals are used to convey meanings. Hurford investigates a learning procedure he calls "Saussurean," where an agent uses its own learned responses to signals to determine what to send, and shows that Saussurean learners often develop highly coordinated signaling systems.

While the complexity of the current model precluded using Hurford's learning procedure directly, the sending mechanism was designed to enforce a relation between an agent's transmission and reception behavior. Since the agent's network processes each character in a sequence as it is sent, the network's units will have identical activation values after processing each character when sending a sequence and when receiving that same sequence. Therefore if it stopped sending according to step s.5.a, when each of its

s.1 Set all network inputs to zero.

s.2 Choose a character to send:

- **s.2.a** For each character, determine how many output values would be correct, were that character processed.
- **s.2.b** Choose the character for which this value is highest.
- **s.2.c** If more than one character would give the same number of correct outputs, choose from them the one whose error is lowest.

s.3 That character is sent to the hearer.

s.4 That character is processed by the network, as described in step h.2 in Figure 3.

s.5 After the character is processed:

- **s.5.a** If each value in the network's output layer is correct, stop sending characters.
- **s.5.b** Otherwise if the total length of the sequence sent exceeds a maximum value, stop sending characters.
- **s.5.c** Otherwise continue at step s.2.

Figure 4: Operation of the recurrent networks shown in Figure 2, when used to send a sequence of characters.

output activation values was correct, the network will, after hearing the sequence, also have the correct meaning. Thus the agent will correctly interpret sequences that it sends. (This will not be true, on the other hand, for sequences terminated because the cutoff length is reached.)

The assumption that agents can use their own responses to characters as a means to predict the responses of others is crucial to the present model, and must be satisfied in any animals to which it could be applied. The ability to use one's own cognitive and emotional responses to potential situations to predict or understand those of others is of great value for animals with rich social interactions. It appears to be present to some degree in the primates, and is relatively developed in apes, though not nearly to the degree it is in humans. (See the papers in Whiten, 1991).

3 An Interpretation of the Meaning Vectors

The meanings transfered between the agents are just vectors of numerical values. The mnemonic interpretation of the meaning vectors described in this section will be used in the analysis of the sequences used to convey them. Although the interpretation is motivated by speculation about the emergence of grammar in humans, and involves a model of properties of natural language pronouns, I am *not* claiming that results of the computational simulations constitute any sort of justification for either. The simulations are intended to explore whether agents can develop coordinated systems for conveying structured meanings. The following interpretation of the meaning vectors is intended only to facilitate the analysis of whether and how they do.

A crucial event in the emergence of grammar occurred when discrete signals, perhaps like those used as alarm calls by vervet monkeys (Cheney and Seyfarth, 1990), began to be treated as being composed of relatively independent meaningful segments that could be systematically combined to produce new signals. For this to happen, their users must have been capable of comprehending meanings whose content could be analyzed into more or less independently meaningful components, perhaps as involving a group of one or more individuals that manifest some property or relation, or are participating in some type of process or event. Given the capacity to comprehend such meanings, it might then have been useful to express them.

In human languages such meanings are expressed with a clause, headed

Predicates

values	
011001	*happy*
011100	*sad*
101001	*angry*
100011	*tired*
110001	*excited*
100101	*sick*
100110	*hungry*
000111	*thirsty*
010101	*silly*
010011	*scared*

Referents

sp	*hr*	*ot*	*pl*	
1	0	0	0	*me*
1	0	0	1	*we*
1	0	1	1	*mip*
0	1	0	0	*you*
0	1	0	1	*yall*
0	1	1	1	*yup*
1	1	0	1	*yumi*
0	0	1	0	*one*
0	0	1	1	*they*
1	1	1	1	*all*

Example Meanings

values	
0010101001	*(one angry)*
1101010101	*(yumi silly)*
1111100101	*(all sick)*
0111100110	*(yup hungry)*
1011010101	*(mip silly)*
0111100101	*(yup sick)*
0100100011	*(you tired)*
0011000111	*(they thirsty)*
1001011100	*(we sad)*
1000110001	*(me excited)*

Figure 5: Meaning vectors for the predicates (left), referents (center), and ten example meanings (right).

by a verb whose inflectional form often conveys information about one or more of its thematic arguments, as well as about the process or situation involving the individuals those arguments refer to. In some languages, the information conveyed by its inflection enables a single verb form to function as a complete utterance.

The interpretation of the meaning vectors is therefore motivated by the possibility that the grammar of an early communication system might have resembled the inflectional morphology of verbs in modern languages. The values in the meaning vectors are partitioned into two groups: Six of the values are taken as encoding a predicate, and the remaining four are taken as encoding a referent that the predicate applies to. There are ten patterns each for the predicates and referents, and therefore 100 different meanings that can be represented. The meaning vectors corresponding to the predicates and referents are shown in Figure 5, in addition to ten example meanings.

The meaning vectors for the predicates are randomly chosen, such that each has three positions whose value is 1 and three whose value is 0. The word assigned to each of them is entirely arbitrary. (For example, there is no intended relationship between the pattern representing *happy* and that representing *sad*.)

As with the set of predicates, the names used for the referent have no

significance other than the vector of values they stand for. However the names are related to the values in the meaning vectors according to a simple model of natural language pronouns. Each position in the referent vector indicates whether a certain property is true of the set of one or more referents. The first position represents whether the set includes the speaker (sp) or not. The second position indicates whether the hearer (hr) is included. The third position indicates whether any other (ot) individuals are included. The fourth position indicates whether the set of referents is plural (pl) or not.

Given this set of properties, there are ten combinations that are logically consistent. The names assigned to them are based on words of English and the English-based Creole language Tok Pisin, spoken in Papua New Guinea. The features of the referents *me*, *you*, and *they* correspond to those of the English pronouns. The referent *one* is third-person singular. *Mip* and *yup* (based on Tok Pisin 'mipela' and 'yupela') refer to groups containing either the speaker or the hearer, respectively, but not both, in addition to at least one other individual. The referent *yumi* (also from Tok Pisin) designates a group containing only the speaker and hearer. The referents *we* and *yall* designate the speakers, or hearers, respectively, when construed as a group. The referent of *all* is a group including the speaker, the hearer, and others.

4 Negotiation of Coordinated Communication

To communicate successfully, the members of a population of agents must all send more or less the same sequence of characters for each of the meanings, and must be able to correctly interpret most of the sequences sent by the other members. Coordinated communication is achieved in the simulation runs as each agent in the population alternates between learning to interpret the sequences sent by others, and sending sequences for others to emulate. I characterize this process as "negotiation" because all of the agents both contribute to, and conform to, the population's communication system as it develops.

A simulation run is begun by creating a population of agents and initializing the connection weights of their networks to random values chosen from a uniform distribution between -0.5 and $+0.5$. The simulation run

n.1 Choose an agent at random from the population to act as the "learner."

n.2 Repeat 10 times:

- **n.2.1** Choose an agent other than the learner at random from the population to act as the "teacher."
- **n.2.2** Train the learner's network to correctly interpret the sequences sent by the teacher, each presented once, in random order.
- **n.2.3** Return the teacher to the population.

n.3 Return the learner to the population.

Figure 6: A negotiation round.

correctness The fraction of communicative episodes observed in which each value in the hearer's meaning vector is correct.

error The average root mean square error between the hearer's meaning vector and that of the speaker after a communication episode.

distinctness The average fraction of sequences an agent sends for exactly one meaning.

length The average total length of the set of sequences sent by each agent for the set of meanings. A value 1.0 indicates that the length of each sequence equals the cutoff value of 20.

Figure 7: Quantitative measures for assessing the degree of coordination of communication in a population. Each value is based on a sample of communicative episodes observed after a round of a simulation run.

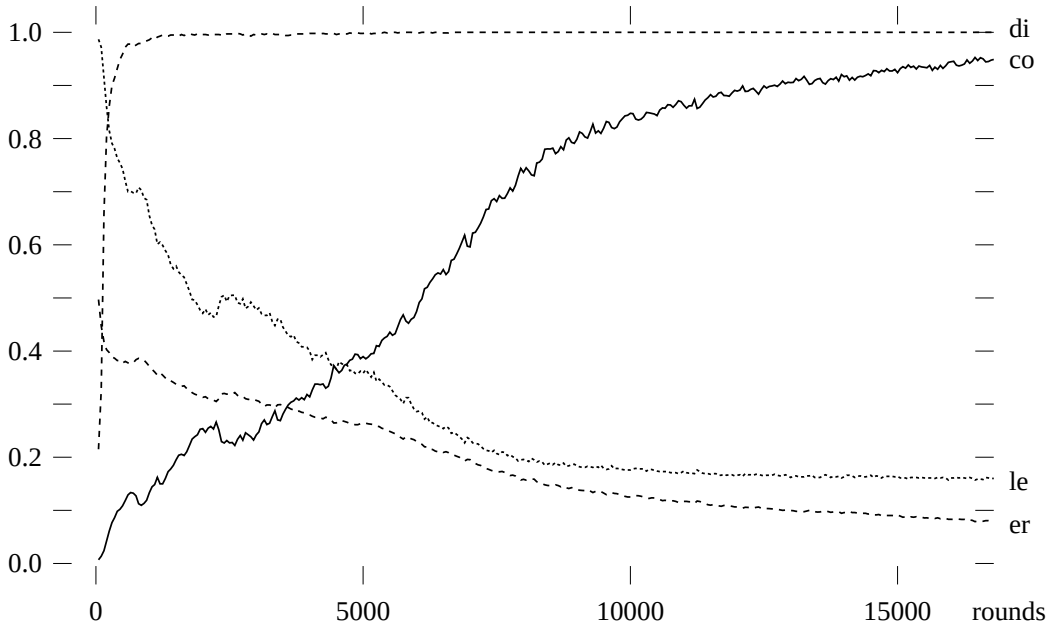

Figure 8: Record of a simulation run of a population of 30 agents. Plotted each round are the population's correctness (co), error (er), distinctness (di), and length (le) values. (See Figure 7.)

then proceeds as a sequence of "rounds" of negotiation, performed as described in Figure 6. Quantitative measures of the degree of coordination of the population's communication, described Figure 7, are recorded after each round of the simulation. Figure 8 presents the record of a simulation run of a population of 30 agents.

In the first round of the simulation illustrated, the randomly initialized agents have a correctness value near 0.0[1], their error is approximately 0.5, their distinctness is low and the average length of their languages is high. All of these values indicate that the members of the initial population do not succeed very well in their communicative attempts; indeed the fact that the average length of the languages is near the maximum indicates that the agents are unable to interpret their own sequences accurately.

[1]Its expected value is 2^{-n}, where n is the number of values in the meaning vectors.

In the early rounds of the negotiation, the distinctness value rises sharply, from near 0.20 at round 1 to above 0.90 by round 300. This happens because each learner observes the sequences sent by ten other agents for each meaning. Its training input is therefore almost certainly different for each meaning, even though no individual agent sends a different sequence for each meaning.

By round 300 each agent sends more than 90 different sequences for the 100 meanings, but the agents are still not very accurately interpreting them. Each learner will see, in general, ten different sequences for each meaning as it is trained. This contradictory input makes it unlikely that after training it will correctly interpret any of them very well. Still, slight statistical fluctuations do occur, and increase the likelihood that certain sequences will be sent for a given meaning. Such fluctuations are amplified as each agent is trained, and the population starts to converge towards agreement about the sequence to be sent for each meaning.

As this agreement develops, the agents are exposed to increasingly less contradictory training input. They are therefore able to learn the developing system with greater accuracy, as shown by the increase in correctness and the decrease in error. The average length of the sequences sent steadily decreases, as an agent need only send enough characters to differentiate one meaning from the others.

By round 15000, the population has achieved a very high level of communicative accuracy. Over 92% of the meanings are being interpreted accurately. The error has dropped to below 0.1, and the agents send different, and short, sequences for each meaning. This run was continued for a total of 35670 rounds, at the end of which 97.6% of the meanings were being interpreted correctly, with an average error of 0.044.

Figure 9 presents the sequences that an agent sends for some of the meanings before, and after, a simulation run. The initialized agent's sequences are all the maximum length, and the agent sends only 34 different sequences for the set of 100 meanings. After the simulation run the network's sequences are significantly shorter, and it sends a different one for each meaning.

Figure 10 illustrates that the agents in a population after a simulation run can interpret each other's sequences accurately. For several of the meanings, the sequence sent by one agent is shown, as well as the meaning vector of another agent after processing the characters in the sequence. Note that almost all of the values are "correct" in the sense of being within 0.5 of the

meaning	*sequence*
(yumi scared)	cccccccccccccccccccc
(mip hungry)	dbdddbdddbbdbdddddbd
(yall sad)	ccbaccccacaadbcabcac
(they scared)	dbddccdcccccccccccc
(mip silly)	aaaaaaaaaaaaaaaaaaaa
(yall angry)	cccccccccccccccccccc
(yup sad)	ddddddddddddddddddddd
(me angry)	aaaaaaaaaaaaaaaaaaaa
(yup silly)	adddddddcddddddddadcd
(we hungry)	ccddccccccdddccccdcc
(they sad)	cccccccccccccccccccc

meaning	*sequence*
(yumi scared)	cacd
(mip hungry)	dbd
(yall sad)	acb
(they scared)	caad
(mip silly)	ada
(yall angry)	bcdbbbbbb
(yup sad)	abac
(me angry)	bdd
(yup silly)	adba
(we hungry)	ddc
(they sad)	abab

Figure 9: Sequences sent by an agent for some of the meanings when its connection weight values are initialized to random values (left), and after it has participated in a simulation run (right).

value in the speaker's meaning vector, but, for the most part, they are much closer than that, consistent with the low error value the population archived.

5 Analyzing the Systems

While the agents can evidently interpret each other's sequences correctly, this ability does not necessarily require that any systematic regularities exist between the meaning patterns and the sequences that convey them. The agents might have just settled on a set of short and distinct, but unrelated, sequences to convey the meanings. In this section I describe some of the languages that emerged in simulation runs and seek to elucidate whatever systematicity they possess.

At the end of the simulation that involved the agents whose performance is illustrated in Figures 9 and 10, the languages produced by each agent in the population were compared. For 65 of the meanings, each agent in the population produced exactly the same sequence. For 14 of the meanings, all of the agents but one produced the same sequence, and for 17 of the meanings, all of the agents but two produced the same sequence. For the remaining meanings, most of the population produced the same sequence,

(me happy)			`ba`						
1	0	0	0	0	1	1	0	0	1
0.97	0.02	0.05	0.19	0.00	1.00	0.99	0.00	0.00	0.98

(yumi scared)			`cacd`						
1	1	0	1	0	1	0	0	1	1
0.98	0.97	0.00	0.98	0.00	0.99	0.00	0.01	0.99	1.00

(you scared)			`caca`						
0	1	0	0	0	1	0	0	1	1
0.00	0.97	0.00	0.01	0.02	0.99	0.00	0.00	0.99	1.00

(mip hungry)			`dbd`						
1	0	1	1	1	0	0	1	1	0
0.97	0.00	0.99	0.92	0.98	0.00	0.00	0.99	0.99	0.02

(yall sad)			`acb`						
0	1	0	1	0	1	1	1	0	0
0.05	1.00	0.00	0.95	0.00	0.99	0.96	0.99	0.00	0.02

(they scared)			`caad`						
0	0	1	1	0	1	0	0	1	1
0.10	0.00	0.99	0.83	0.00	0.97	0.00	0.00	1.00	1.00

(mip silly)			`ada`						
1	0	1	1	0	1	0	1	0	1
0.49	0.07	0.99	0.92	0.00	0.99	0.00	0.99	0.00	0.99

(yall angry)			`bcdbbbbbb`						
0	1	0	1	1	0	1	0	0	1
0.26	1.00	0.03	1.00	0.99	0.00	0.99	0.00	0.00	1.00

Figure 10: One agent interprets sequences sent by another. Two agents that participated in a simulation run were chosen, one to act as the speaker, the other as the hearer. Each box above shows, for one of the meanings, the sequence sent by the speaker. Below that is that is shown the correct pattern of 1's and 0's corresponding to that meaning. The last line in each box shows the hearer's meaning vector after processing the sequence.

though there were more alternatives produced, usually by one or two agents each.

5.1 A Paradigmatic Analysis

The sequences sent by a majority of the population for each of the meanings are arranged as a paradigm, in accord with the motivation behind the interpretation of the meaning vectors, at the top of Figure 11. The agent whose sequences are shown in Figure 9 and 10, as it happens, doesn't follow the majority exactly, most strikingly in the sequence it sends for *(yall angry)*.

While not completely regular, the sequences do exhibit some systematicity. A quasi-linguistic analysis of the system is shown at the bottom of Figure 11. Each sequence is analyzed as a root that expresses the predicate, plus some modification to the root that expresses the referent. The analysis is illustrated by replacing the characters of the supposed root with the symbol '`-`', followed by, or interspersed with, the modifier characters. The predicates are ordered so that the most regular part of the paradigm is at the top.

For the predicates *tired*, *scared* and *sick*, all of the sequences can be analyzed as a root plus a suffix that determines the referent. For the referent *me* the suffix is empty.

Sequences expressing the predicate *happy* do not completely conform to this pattern. Instead of adding the character `c` to the end of the root form to express the referent *you*, the character `c` is inserted between the two characters of the root. A similar internal change, also with `c`, occurs when the referent is *yall*. Instead of ending with `ba` for the pronoun *yup*, the suffix `ac` appears. For *we* and *all*, sequences whose predicate is *happy* add an extra `c`, compared with the first three predicates; and for *mip* and `yumi`, the suffix does not include the final `d` that sequences for the first three predicates use.

The sequences for *sad* and *excited* also deviate from the regularities exhibited by the first three predicates, but at least some of the differences might be due to the fact that the root forms for these two referents consists of a single character, as opposed to two for the predicates above them. The root is followed by `b` for the pronouns *one*, and *they*, but otherwise the sequences for those referents are consistent with the predicates above them. For *you* and *yall* the suffixes are the same as the first three predicates, except that sequences whose predicate is *excited* have an `a` following the initial `c` in the

	one	*they*	*you*	*yall*	*yup*	*me*	*we*	*mip*	*yumi*	*all*
tired	cda	cdab	cdc	cdcb	cdba	cd	cdd	cddb	cdcd	cdb
scared	caa	caab	cac	cacb	caba	ca	cad	cadb	cacd	cab
sick	daa	daab	dac	dacb	daba	da	dad	dadb	dacd	dab
happy	baa	baab	bca	bcab	baac	ba	badc	bab	bac	babc
sad	aba	abab	ac	acb	abac	a	abdc	abb	abc	abbc
excited	cba	cbab	cca	ccab	cbca	c	ccdc	cb	ccb	cbc
angry	bb	bbb	bc	bcb	bbc	b	bddc	bdb	bdc	bdbc
silly	aa	aaab	aca	acab	adba	add	addc	adad	adc	adbc
thirsty	dbaa	dbab	dca	dcba	dbca	dda	ddac	dbad	dcad	dbacd
hungry	dbb	dbbd	dc	dcb	dbc	dd	ddc	dbd	dcd	dbcd

	one	*they*	*you*	*yall*	*yup*	*me*	*we*	*mip*	*yumi*	*all*
tired	--a	--ab	--c	--cb	--ba	--	--d	--db	--cd	--b
scared	--a	--ab	--c	--cb	--ba	--	--d	--db	--cd	--b
sick	--a	--ab	--c	--cb	--ba	--	--d	--db	--cd	--b
happy	--a	--ab	-c-	-c-b	--ac	--	--dc	--b	--c	--bc
sad	-ba	-bab	-c	-cb	-bac	-	-bdc	-bb	-bc	-bbc
excited	-ba	-bab	-ca	-cab	-bca	-	-cdc	-b	-cb	-bc
angry	-b	-bb	-c	-cb	-bc	-	-ddc	-db	-dc	-dbc
silly	(aa)	(aaab)	(aca)	(acab)	--ba	--d	--dc	--ad	--c	--bc
thirsty	-b-a	-b-b	-c-	-cb-	-bc-	-d-	-d-c	-b-d	-c-d	-b-cd
hungry	-bb	-bbd	-c	-cb	-bc	-d	-dc	-bd	-cd	-bcd

Figure 11: Sequences used by a majority the population for each of the given meanings (top). A potential analysis of the system in terms of a root plus modifications (bottom). Sequences in parentheses cannot be made to fit into this analysis.

suffix. The rest of the paradigms for these two predicates differ substantially from those of the first three predicates shown, but seem to be more similar to the one for *happy*, and this similarity continues to the sequences for the predicate *angry*, whose root form also has one character.

The first four entries in the paradigm for *silly* cannot be easily analyzed as a root form with modifications as with the predicates above it. In particular, the sequence for *me* is certainly not the bare root. Still, many of the entries in this row are similar to those of the other predicates.

Sequences for *thirsty* and *hungry*, while also showing some similarities to those of the other predicates, don't completely conform to any of their patterns.

5.2 Shared Trajectories Through Activation Space

Like any linguistic analysis, the one just outlined would open to question on a number of issues were it proposed to account for the inflectional system of a newly-described human language. However the analyses would be at least provisionally justified by the fact that similar analysis techniques are known to apply to other human languages, and seem to express deep regularities among them. Of course this is not a human language, and so the plausibility of the analysis has no such support. The existence of partial regularities in the system can be used as a convenient way to group the sequences, but may be entirely artifactual, and certainly does not entail that any sort of analysis in terms of those regularities is involved in the agents' interpretation of the sequences.

A more plausible account, more consistent with the operation of recurrent neural networks, is that characters and short sequences encode trajectories through the vector space of network activation values. To express each of the meanings with a distinct sequence, and to be able to correctly interpret sequences, the weights of an agent's network must be such that for each meaning there is a sequence of characters that moves the network's output activations through a trajectory to that meaning vector.

For the members of a population of agents to communicate successfully, the trajectory followed by each agent's network output activations on an identical sequence must be roughly similar. If they were to diverge substantially at some point in the sequence, there would be little chance that the sequence of remaining characters would bring them back together, such that

they will both end up at the same vector of output values.

It is therefore likely that the systematicity observed in the communication systems that emerge in the simulation runs is due to the networks acquiring a shared set of mappings from partial sequences to transformations of their output values. If this set of transformations can be composed to span the set of meanings, the agents will be able to assemble sequences of characters that will guide each other's networks through trajectories in activation space that will terminate in the correct meaning vectors.

This explanation is consistent with the sort of partial regularity seen in the negotiated systems. For example in the system shown in Figure 11, the supposed roots for the top predicates guide the agents output values to vectors corresponding to the meaning corresponding to that predicate applied to the referent *me*. The subsequences corresponding to the suffixes then move the activation values slightly, changing only the values corresponding to the referent.

On the other hand, this is not the only way that the output activation values of the network can be guided to their correct values. Other partial sequences will also move the activation values along specific paths, and the agents will use them if the regular sequence doesn't quite work. For example the sequences `bad` and `bab` would have fit more consistently into the system shown in Figure 11 to express express *(we happy)* and *(all happy)*, respectively. Apparently the former sequence does not quite bring the output activation vector to the correct values, and so an additional `c` is added, indeed this final `c` is used for all of the entries in the column except the three supposedly regular entries. The sequence `bab` can't be used to express *(all happy)* because it is used for *(mip happy)*. The additional `c` is needed to adjust the output vector to add a 1 in the position corresponding to the *ot* (*other*) feature. (See Figure 3.) The *ot* feature also needs to be 1 in meanings whose referent is *we*, and this seems to be the effect of the final `c` in most of those entries.

It also possible that specific sequences used to guide the network activation values to correct values might be used for only a small set of the meanings, or that meanings involving different predicates or referents would use different sets of sequences. For example the sequence `aa` is used to express *(one silly)*, and while this can't be seen as conforming to the "root plus suffix" analysis in much of the rest of the paradigm, the sequence `ab` can still be used, as it is for other predicates, to adjust the meaning vector to express

the referent *they*. In the entries for the referent *yup*, five of the predicates end with `ba`, and four of the remaining predicates use a sequence that includes `bc`, indicating that at least two routes to this predicate are available to the agents. About half half of the predicates use one, and most of the rest use the other.

This explanation also accounts for one aspect of the systems that emerge in the simulations that is certainly an artifact, namely that most of them, including the one just discussed, tend to express the predicate portion of the meaning with the first few characters of the sequences. Recall from Section 3 the predicate involves six of the values in the meaning vector with the remaining four used to express the predicate. Since the networks generate sequences by determining which character would bring their outputs closest to the meaning being expressed, as determined by counting the number of correct values, characters that modify the output values to get the predicate correct will tend to be chosen. Even so, a few systems have emerged in which the referent is expressed first. Simulations are currently being performed in which predicates and referents are expressed with equal numbers of meaning vector values.

5.3 Conveying Novel Meaning Combinations

To determine whether the regularities in a population's communication system enable members of that population to convey novel meaning combinations, a simulation run was performed in which ten of the meanings were omitted. The communication system that emerged in this simulation is shown in Figure 12, with blank spaces indicating the omitted meanings.

After the simulation run was complete, one of the agents from the population was used to generate sequences for each of the omitted meanings, and another agent was used to interpret them. The results are shown at the bottom of Figure 12.

Even though the agents have never sent sequences for these meanings before, nor have they ever (correctly) interpreted a sequence as one of them, the agents are able to convey them with reasonably good accuracy, using sequences that seem to obey the regularities of the negotiated system. The fact that several of the sequences used for the new meanings are longer than the entries in the paradigm supports the idea that that the systematicity observed in the systems is due to the agents making use of a shared set

	one	*they*	*you*	*yall*	*yup*	*me*	*we*	*mip*	*yumi*	*all*
happy		dbad	dac	dcc	bba	dab	dcb	dbd	dbc	bbad
sad	dada		da	dca	dda	dad	dcd	ddd	dd	ddad
angry	abba	cbba		cba	bbba	abb	cbb	bbbd	bbcb	bbba
tired	aa	ca	ac		ba	acd	ccd	bcd	bc	bad
excited	aab	cab	ab	cbc		abd	cbd	bbd	bbcd	bbda
sick	aabb	cabb	acb	ccb	bab		ccbd	bcbd	bcb	babd
hungry	aacb	cacb	accb	ccc	bacb	acdb		bcdb	bccb	bacd
thirsty	aaa	caa	aca	cca	baad	acad	ccda		bdca	badad
silly	aaab	caab	ada	cda	baaa	adb	cdb	bdbd		baabd
scared	aad	cad	ad	cd	baac	add	cdd	bdd	bd	

(yall tired)			cc						
0	1	0	1	1	0	0	0	1	1
0.01	0.98	0.00	0.99	0.88	0.00	0.00	0.44	0.99	0.92

(yup excited)			b̂bdaccca						
0	1	1	1	1	1	0	0	0	1
0.01	1.00	0.97	1.00	0.99	0.53	0.00	0.06	0.00	1.00

(me sick)			abdbd						
1	0	0	0	1	0	0	1	0	1
0.99	0.02	0.00	0.00	0.99	0.17	0.22	0.06	0.00	0.99

(we hungry)			cccdbdc						
1	0	0	1	1	0	0	1	1	0
0.91	0.00	0.15	1.00	0.99	0.00	0.00	0.99	1.00	0.18

(mip thirsty)			bddba						
1	0	1	1	0	0	0	1	1	1
1.00	0.29	1.00	1.00	0.00	0.00	0.00	0.76	0.98	0.95

(yumi silly)			bdaa						
1	1	0	1	0	1	0	1	0	1
1.00	1.00	0.03	1.00	0.00	0.70	0.23	0.99	0.00	1.00

Figure 12: Negotiated communication system of a population for 90 of the meanings (top). Blank spaces in the paradigm indicate meanings not used in the negotiation. Sequences sent by a member of the population and the meaning vectors of another member of the population when shown those sequences, for some of the meanings left out of the original negotiation (bottom).

of mappings from partial sequences to trajectories in the space of output vectors. Since the agents have never conveyed these meanings before, the mappings they have acquired do not enable then to guide each other's outputs directly to the desired values in all cases. But since the mappings that they possess are shared, the agents are able to compose longer sequences that are correctly interpreted.

6 Conclusions

The negotiation model used in the simulations was arrived at after a number of different approaches, based on evolutionary simulations in which the reproductive fitness of agents depended on their communicative accuracy, failed to achieve anything like the results described above. While the precise reasons for these failures, and their significance, is unclear, the fact that complex adaptive coordination can emerge from social interactions among a population of agents is an important lesson, independent of any specific relevance to language. The coordination is achieved in the simulations as a result of a distributed process in which individuals learn by observing the behavior of others, with no external guidance over the how the system ought to develop.

I have argued elsewhere (Batali, 1993) that other representational and Intentional phenomena can be understood as the result of processes of social activity among populations of animals, whose details are influenced by the animals' cognitive abilities, by the external media in which their interactions occur, and often by arbitrary historical contingencies. Such processes leave enduring traces, for example as modifications to the external environment, or as systematic regularities in the animals' behavior, that can then become a cognitive resource for the animals, enabling even more coordination of their activity.

The most crucial assumption underlying the model involves the procedure used to send sequences. As discussed in Section 2, the procedure requires that an animal use its own cognitive responses to predict those of others. While such abilities are rare among animals, the fact that existing hominoids possess them suggests that early hominids did too. The increasing complexity of social organization seen during hominid evolution was most likely accom-

panied by enhancement of all of the cognitive abilities that underlie social activity.

As the hominids began to develop the abilities to use their own responses to predict and influence those of others, situations might have arisen in which such abilities could have been used to communicate information. With the support of shared context, such communicative attempts could often have been successful, even without without any coordinated system.

If they were capable of learning from each other's communicative behavior, however, the simulations described in this paper suggest that early hominids could develop systems to express structured meanings without any innate language-specific traits. Their communicative behavior would have exhibited systematic regularities from which some of the grammatical resources of modern human languages might have emerged.

7 References

John Batali. Trails as archetypes of Intentionality. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, pages 220–225, 1993.

John Batali. Innate biases and critical periods: Combining evolution and learning in the acquisition of syntax. In Rodney Brooks and Pattie Maes, editors, *Proceedings of the Fourth Artificial Life Workshop*, pages 160–171, Cambridge, MA, 1994. The MIT Press.

Dorothy L. Cheney and Robert M. Seyfarth. *How Monkeys See the World: Inside the Mind of Another Species.* University of Chicago Press, 1990.

A. Cleeremans, D. Servan-Schreiber, and J. L. McClelland. Finite state automata and simple recurrent networks. *Neural Computation*, 1:372–381, 1989.

P. D. Eimas, E. R. Siqueland, P. Juscyk, and J. Vigorito. Speech perception in infants. *Science*, 171:303–306, 1971.

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

Simon Haykin. *Neural Networks: A Comprehensive Foundation.* Macmillan, New York, 1994.

J. R. Hurford. Biological evolution of the Saussurean sign as a component of the language acquisition device. *Lingua*, 77:187–222, 1989.

Michael Jordan. Serial order: a parallel distributed processing approach. Technical Report ICS Report No. 8604, Institute for Cognitive Science; University of California at San Diego, 1986.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing*, volume 1. MIT Press, Cambridge, MA, 1986.

H. T. Siegelmann. *Foundations of Recurrent Neural Networks.* PhD thesis, Rutgers University, Graduate Program in Computer Science, 1993.

Andrew Whiten, editor. *Natural Theories of Mind: Evolution, Development and Simulation of Everyday Mindreading.* Basil Blackwell, Oxford, 1991.