Learning a Common Language Through an Emergent Interaction Topology

Samarth Swarup
Dept. of Computer Science,
University of Illinois at
Urbana-Champaign,
Urbana, IL, USA
swarup@uiuc.edu

Kiran Lakkaraju
Dept. of Computer Science,
University of Illinois at
Urbana-Champaign,
Urbana, IL, USA
klakkara@uiuc.edu

Les Gasser
Graduate School of Library
and Information Science, and
Dept. of Computer Science,
University of Illinois at
Urbana-Champaign,
Urbana, IL, USA
gasser@uiuc.edu

ABSTRACT

We study the effects of various *emergent* topologies of interaction on the rate of language convergence in a population of communicating agents. The agents generate, parse, and learn the meanings of sentences from each other using recurrent neural networks. An agent chooses another agent to interact with and learn from, based on the second agent's *fitness.* Fitness is defined to include a frequency-dependent term capturing the approximate number of interactions an agent has had with others—its "popularity" as a linguistic partner, which in turn is a measure of the proportion of the population which it has taught. This method of frequency-dependent selection is based on our earlier *Noisy Preferential Attachment* algorithm, which has been shown to produce various network topologies, including scale-free and small-world networks. We show that convergence occurs much more quickly with this strategy than it does for uniformly random interactions. In addition, this strategy more closely represents choice preference dynamics in large natural populations, and so may be more realistic as a model for adaptive language.

1. INTRODUCTION

One of the main problems in language evolution is how to get a population of agents to converge to a common language, without globally imposing some kind of hierarchy on the population. Interactions between agents can be described by an *agent interaction matrix* where entry (i, j) in the matrix is the probability that agent i will receive a message (and thus potentially learn) from agent j. Such a matrix captures, in a general way, topological constraints on agent interactions, e.g., spatial locality constraints, constraints based on agents' knowledge of each others' existence, or interaction choice preferences.

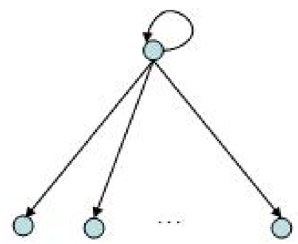

Figure 1: The interaction graph for quickest convergence. One agent teaches the language to all the other agents. The circles represent agents, and the directed arrows represent the influence of one agent on the language of the other agent.

The problem of convergence to a common language is also interesting because it can be seen as a specific instance of a general distributed search and convergence problem in which a set of agents explores a solution landscape with the aim of collectively converging to a state of high objective quality. The "fitness" of a state in this formulation is defined by both an intrinsic measure (the objective quality of the state), and by a frequency-dependent measure (the number of agents that are in that state). Thus the general problem can be stated as "Given a landscape of states and possible moves between them, and a search strategy under which agents learn from encounters with others, what constraints on the agent interaction matrix will ensure the most rapid convergence to some highest-fitness state?" In effect, how should agents make choices of learning partners to achieve the fastest collective convergence to the best objective state? This is the problem we study, in the context of developing a common and high-quality language. The realism, interestingness, and difficulty of this problem is enhanced if we assume that objective quality of states (i.e., languages) can vary over time. For example, as collective tasks change, agents may need to converge to new languages that best communicate new task features. This requires that agents collectively adapt their languages over time as a group, tracking objective quality variations.

The agent interaction matrix can be seen as a weighted directed *influence* graph which describes the influence of an agent on the language of another agent. A simple strategy for rapid convergence would be to designate a special agent from which each the other agent learns its own language, as in figure 1. This corresponds essentially to a pre-imposed or designed language, which may or may not be of the highest objective quality. Such a centralized system is brittle in practice because a) the teacher agent has to be responsible for adapting the language to keep up with changing tasks, environments and needs of all the agents, b) communicative load on the teacher increases at least linearly with population size, reducing scalability, and c) the centralized teacher is a single failure point. Multi-agent systems are generally distributed and open, which means that there is no central control point, and agents may enter and leave the population at any time. This means that although desirable for its speed, uniformity, and certainty, the interaction topology shown in figure 1 is both undesirable and unrealistic for a general multi-agent system.

The rest of this paper is organized as follows. We first describe some recent work investigating the role of the interaction topology in the convergence of language and social conventions in multi-agent systems. Then we describe our experimental framework, which includes a mechanism for the distributed generation of network topologies. This is followed by some experiments and a discussion of the results. Finally we discuss the possibilities for expanding on this work to include situatedness and theoretical modeling.

2. RELATED WORK

There has been significant work on the convergence of a population of agents to a particular language. Work by Cucker et. al.[4] provides a theoretical basis for this problem. Komarova et. al, and Lee et al.[8] study the problem from the point of view of population dynamics, while Shoham and Tennenholtz[10] and Delgado[5] study the general problem of emerging social conventions.

Currently available work, such as that of Delgado and Lee et al., does investigate linguistic convergence in non-uniform interaction regimes. However, this prior work has assumed that interaction topologies are fixed, not emergent. Additionally, the language- or convention-forming agents studied in this work have not been collective learners who are exploring a wide space of alternatives; they are simply choosing among two options.

Cucker et al.[4] ("CSZ") recently presented a result wherein they showed that an interaction matrix must be *weakly irreducible* in order to ensure convergence. An interaction matrix, which is a non-negative square matrix, is said to be weakly irreducible when the corresponding stochastic matrix (obtained by normalizing each row to sum to 1) is weakly irreducible. A stochastic matrix is said to be weakly irreducible if 1 is a simple eigenvalue and all its other eigenvalues are less than 1 in modulus. Weak irreducibility essentially ensures the existence of sufficiently many non-zero elements in the matrix to prevent the population from getting partitioned into disjoint non-communicating subgroups.

The speed of convergence in the CSZ setting depends on a number, α_*, which is the second-largest eigenvalue of the interaction matrix (the largest eigenvalue is 1). They define a metric, d, which measures the distance from the current set of languages in the population to a state where every agent speaks (almost) the same language. They show that if each agent sees a sufficient number of examples at each time step, then there exists a constant α, $\alpha_* < \alpha < 1$, such that with probability $1 - \delta$,

$$d(t) \leq \alpha^t d(0). \tag{1}$$

Shoham and Tennenholtz [10] examined the coordination of social conventions among agents. In this model, agents play two-choice games pairwise with the others in the population and make changes to their behavior based on these interactions. The choce of interaction partners is made uniformly randomly with any other agent in the population.

Delgado[5] extended the Shoham and Tennenholtz work and examined the emergence of social conventions in using essentially the same two-choice pairwise game model but varying the possible interaction topologies [5]. He showed that small-world or scale-free topologies are much more efficient than regular topologies (e.g., where every agent interacts with its neighbors) with the same number of average interactions per agent. Scale-free interaction graphs made the system as efficient for convergence as fully-connected graphs. However, in Delgado's work, the topology of interaction was pre-imposed by a central controller, and not influenced by agent choice. Thus, while the peformance results are interesting, the approach provides neither a strategy for autonomous choice of interaction partners nor a model for emergent interaction patterns. Further, there was no learning in the Delgado experiments.

Lee et. al [8] studied the role of the interaction topology on the convergence of a populatin to a single language. This study looked at a set of specific interaction topologies, incuding fully connected, linear, von Neumannn lattice, and a bridge topology. Using the model of Komarova et. al, [7] they empirically studied the critical learning fidelity threshold for language convergence in the various topologies. Although several different interaction topologies were used, the topologies were not emergent and were specified beforehand by the creator of the experiments. In addition, the agents did not learn a language from interactions with other agents, but rather neighbors of high fitness agents were transformed into copies of the high fitness agent with some probability.

We have studied the effects of the topology of interaction on a population of language learners. Our agents use recurrent neural networks to generate, parse, and learn languages from each other. Further, the interaction topology emerges in a distributed manner as a consequence of a specific rational agent strategy, based on our own Noisy Preferential Attachment (NPA) algorithm [11], described in the next section. Adding a learning strategy (recurrent networks that are are trained by the examples presented to them through communicative acts), and a strategy for *choosing* teachers adaptively (NPA), together capture the desirable features of *agent autonomy* (learning and choice), *realism* (*emergent* global interaction patterns), and *flexibility* (continuous adaptation of the collective language space). Finally, having

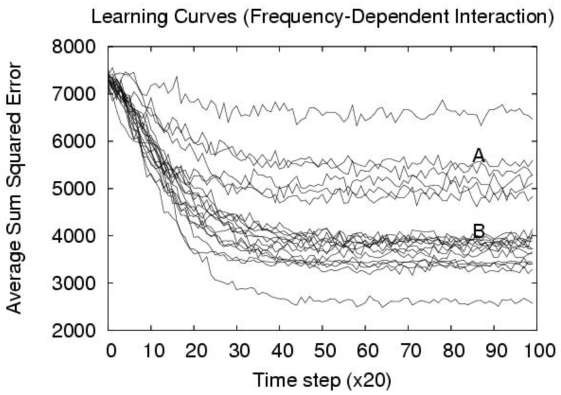

Figure 2: The learning curves for frequency-dependent selection. The curves are divided into two major clusters, marked A and B, with two outliers.

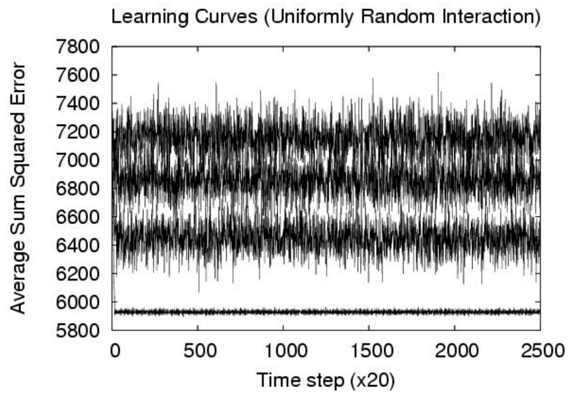

Figure 3: The learning curves for uniformly random interaction. Though we ran the experiment twenty times, only four of the curves are shown. All the other runs overlap with the top 3 curves. In only one case (the lowest curve in the figure) did we see any reduction in error.

an explicit intelligent strategy for choosing teachers grounds the emergent, possibly non-uniform interaction topology in actual agent choice and behavior, giving a clear link between collective and internal ("macro" and "micro") agent dynamics, which has not often been attended to, to date.

3. EXPERIMENTAL FRAMEWORK

We describe below an algorithm for distributed language learning where the topology emerges from the manner in which agents use fitness to select other agents with whom to interact.

Representation: Each agent's internal representation of a language is a simple recurrent neural network (SRN) [6]. An agent uses the same SRN for both generating and understanding (i.e. parsing) sentences. Symbols are encoded using a 1-of-n encoding. Each SRN has as many input and output nodes as there are terminals in the possible languages. To generate a sentence from an SRN, we initialize its internal state (i.e. the context inputs) to 0.5, and give it an input vector that is set to uniformly random values in the range $[0, 1]$. The output of the SRN is interpreted as a symbol by taking the maximum over the outputs. For example, if output 2 has the maximum value, then the symbol corresponding to output 2 is taken to be the output of the SRN. The output vector is then fed back to the SRN as the input for the next time step, and this process is repeated for a randomly chosen number of time steps. The resulting string is considered to be a sentence in the language of the agent. To generate a new sentence, the "state is cleared" again, by resetting the context inputs to 0.5.

Fitness: Each agent has a fitness, defined as follows:

$$f_i(t) = \alpha L_i(t) + (1 - \alpha) K_i(t). \tag{2}$$

$L_i(t)$ is the intrinsic fitness of the language used by agent i at time t, i.e. the quality of its language, which may correspond to its expressivity or how good it is at describing a task. The intrinsic fitness of a language is time-independent, however the language used by the agent changes over time, and therefore L_i is indicated as being time-dependent. $K_i(t)$ is the number of times agent i has communicated a set of messages to another agent up to time-step t. $K_i(t)$ is normalized over the population to keep it from dominating fitness. α is a constant representing a tradeoff between these two components of fitness.

Interaction topology: At each time-step, an agent chooses another agent with whom to interact, with probability proportional to the fitness of that agent. With a small probability p, however, it might switch to another agent chosen uniformly randomly. This "noise" is helpful in keeping the topology from getting too centralized.

The intuition is that if a large number of agents are listening to a particular agent, then it makes sense to pay more attention to that particular agent because its language is probably shared by many agents.

The parameter p also models incomplete knowledge of the interaction history of the entire population. The assumption is that, since there is no central controller telling each agent the fitness of all the other agents, each agent must maintain an estimate of the fitness of the other agents. Since the agents may not be able to observe all the other agents all the time, this estimate will have some error. It turns out that we can take advantage of this error to produce scale-free topologies. In the absence of error, i.e. if $p = 0$, frequency-dependent selection would result in a pin-wheel topology, as shown in figure 1. This is because initially none of the nodes in the graph have any links, and so the first node to acquire a link acquires all the links. However, when $p \neq 0$, but is small, it can be shown that the resulting topologies have power-law like or exponential degree distributions.

Noisy Preferential Attachment: This model for determining interactions is derived from a version of the preferential attachment algorithm that is based on the replicator-mutator equation [9, 3]. Preferential attachment is a well-known algorithm for generating small-world networks [1]. It is a special case of the replicator-mutator equation, and our version of the algorithm, called *Noisy Preferential At-*

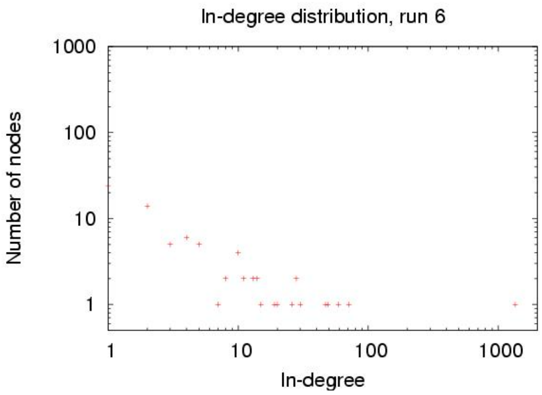

Figure 4: The in-degree distribution for a typical network from cluster B (see figure 2). It is dominated by a single node of high degree, and has a high network betweenness centrality score.

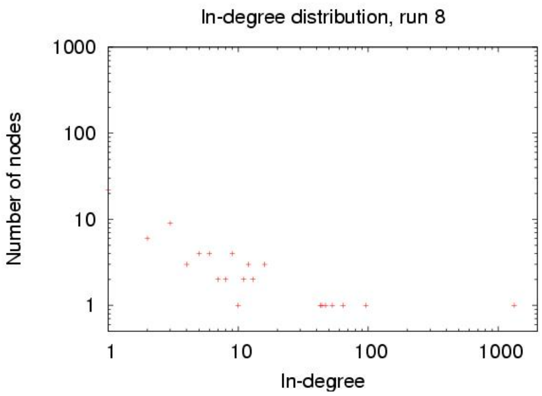

Figure 5: The in-degree distribution for a typical network from cluster A (see figure 2). It is also dominated by a single node of high degree, and has a high network betweenness centrality score.

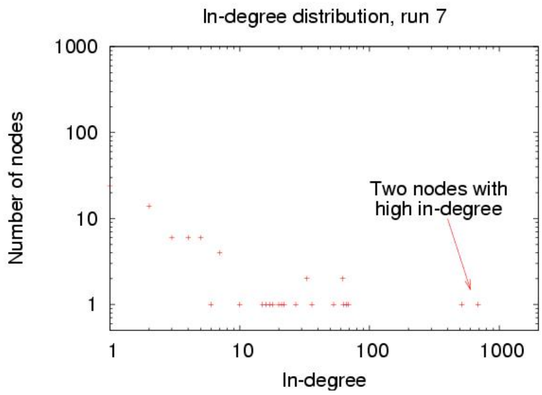

Figure 6: The in-degree distribution for the high outlier. We can see that there are two nodes with a very high in-degree, and the population converges to two distinct languages.

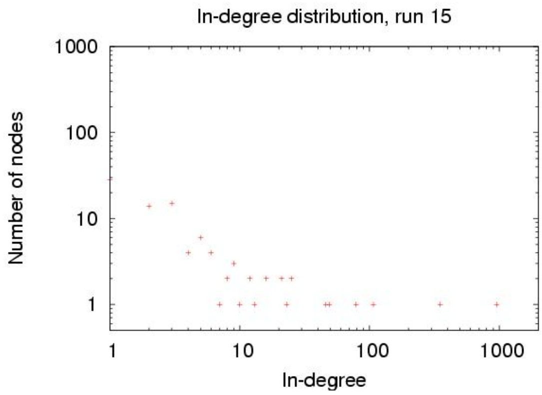

Figure 7: The in-degree distribution for the low outlier, i.e. the run with the best performance. Here also we see that there are multiple nodes of high degree.

tachment [11], can generate random graphs, graphs with exponential degree distributions, and graphs with scale-free degree distributions by varying the parameter p. Thus it provides a convenient framework for testing the effect of the interaction topology on language convergence. In our simulations, the population has a limited memory, and only uses a finite number of the most recent interactions to determine the next one. This is controlled by a parameter called *history*, which defines the window-size to consider when generating the probability distribution for picking the next teacher agent. This overcomes finiteness effects due to the fixed number of agents, and generates scale-free and exponential degree distributions for the interaction graphs.

Experimental paradigm: At each time-step, an agent is chosen uniformly randomly as the learner. It samples m other agents, with replacement, according to the Noisy Preferential Attachment algorithm and gets a set of strings from the language of each agent. Since sampling is done with replacement, agents may contribute multiple sets of strings to this training set. It then updates its own recurrent neural network using this training set. Convergence of the population is estimated after a certain number of time steps by taking a sample of strings from all the agents, and evaluating the parsability of this set for each agent. We measure the average error,

$$E = \frac{1}{n}\sum_{i=1}^{n} e_i, \tag{3}$$

where e_i is the sum-squared error of agent i over the testing set. As the population converges, the value of E decreases.

4. EXPERIMENTS AND RESULTS

We did an experiment with a population of 200 agents. The intrinsic fitness term for each agent, L_i, was set to zero, because in the current experiments the agents are not situated, i.e. they are not performing a particular task, nor are they receiving any external inputs except communication from other agents.

The parameter p, which is the probability of choosing a partner uniformly randomly, was set to 0.05. The maximum sentence length was set to 10. The number of agents sampled by the learner, m, was set to 1.

The set of terminals for all the languages was the same, and consisted of five symbols: `a, b, c, d, e`. Each agent's SRN had five input nodes, six hidden layer nodes, and five output nodes. The symbols were encoded using a 1-of-n encoding, and all weights were initialized uniformly randomly in the range $[-0.5, 0.5]$.

The simulation was run for 2000 time steps, and coherence of the population was evaluated every 20 time steps. The *history* parameter was set to 300 time steps, i.e. an agent only took the last 300 time steps into account when selecting an agent to interact with. The average error of the population when learning occurs with frequency-dependent selection of partners was compared to the case where partners are chosen uniformly randomly. The simulation was run twenty times with identical parameters, and the results are shown in figures 2 and 3. Since no convergence was observed in the uniformly random selection case, we ran it for much longer. However, even after $50,000$ time steps, there was no reduction in error, except in a single case.

The learning curves in figure 2 can be split up into four groups. Clusters A and B, which are marked in figure 2, and two outliers - one high and one low. An examination of the structure of the interaction networks in the frequency-dependent selection case shows something interesting. In-degree distributions for two representative networks from clusters A and B, as well as for the high and low outliers are shown in figures 4, 5, 6, and 7. If agent x learns from agent y at a particular time step, a link is added from node x to node y in the interaction graph.

As expected from the Noisy Preferential Attachment algorithm, all these degree distributions are power-law like. As has been shown by previous researchers [5], populations with a small-world or a scale-free structure of interaction (which are signified by a power-law like degree distribution) tend to be efficient at converging to a common convention. What we see here is that in a more complex situation, like learning a language, the topology can have subtle effects on the convergence.

A key difference between these networks is that the networks in clusters A and B, for which the degree distributions are shown in figure 4 and 5, are dominated by a single node, whereas the outliers both have multiple nodes with high in-degree. This difference can be objectively measured by calculating the *betweenness centrality* for the networks. Betweenness centrality is a value that measures how close a network is to a pin-wheel structure, such as that shown in figure 1. It is a number between 0 and 1, and a low value means that the network is closer in topology to a ring or a fully-connected network. The betweenness centrality values for each of the four classes of networks are shown in table 1. We see that the average betweenness centrality for clusters A and B is very close to 1, whereas the value for the outliers is much smaller.

The surprising fact here is that betweenness centrality is smaller for both the high and the low outlier. This can be explained by examining the actual languages to which the populations converge.

Network	Average betweenness centrality
High outlier	0.5451
Cluster A	0.9348
Cluster B	0.9016
Low outlier	0.7881

Table 1: The betweenness centrality for a network measures to what extent it is dominated by a single node. Networks in clusters A and B are dominated by a single node, as can also be seen from the degree distribution plots in figures 4 and 5.

4.1 Language Accuracy Results

In addition to evaluating the sum squared error on the evaluation set, we measured the similarity of languages between agents at the end of the training regime.

Simple recurrent networks that are initialized randomly tend to have point attractors. This means that when we generate a string of symbols, we observe some initial transient behavior, followed by a single repeated symbol. Thus a typical sequence that is generated might be `b c a a a ...`. The initial `b c` is the transient, after which the SRN falls into the `a`-attractor. Very few randomly initialized SRNs will exhibit more complex languages. An example of a more complex language is a cyclic attractor, which might generate a string like `c b b a a b b a a ...`, i.e. some initial transient followed by `b`'s and `a`'s repeated and alternating. It is much easier to learn a simple language, corresponding to a point attractor, than to learn a more complex language with an SRN, even though SRNs are capable of learning some context-free and context-sensitive languages [2]. This fact, coupled with the low initial frequency of complex languages, led to the dominance of simple languages in all the runs, after convergence. Thus we evaluated the languages after convergence in the following way.

For the last evaluation set of each run, we looked at the sample sentences generated by each agent. For each set of sample sentences, the distribution of letters was calculated. The distribution of letters is a normalized measure of the frequency of each letter in the sentences generated by an agent. For instance, if an agent generated 5 sentences, where three of the sentences were strings of `a`'s of length 5, and two of the sentences were strings of `c`'s of length 3 and 4 respectively, the letter distribution would be $\frac{15}{22}$ `a`'s, 0 `b`'s, $\frac{7}{22}$ `c`'s, 0 `d`'s, and 0 `e`'s (assuming an alphabet of `a, b, c, d, e`). The distribution is an empirical probability distribution of the occurrence of the letters based on the agents sentences.

An agents language is *x letter dominated* if the empirical probability of the letter `x` is greater than the *letter dominance threshold*. When the letter dominance threshold is greater than 0.5 the agent's language will either be uniquely letter dominated by some letter `x` or else will not be letter dominated.

For the entire set of agents, we calculated the number of agent languages that were `a` letter dominated, `b` letter dom-

	a	b	c	d	e	none	mean frequency	variance
Freq-dep Interaction	5	0	13	1	0	1	.7715	.008
– Cluster A	1	0	5	0	0	0	.644	.005
– Cluster B	4	0	8	1	0	0	.811	.001
Uniform Interaction	0	0	0	0	0	20	nan	nan

Table 2: The dominating language in the 20 frequency-dependent interaction runs and the 20 uniformly random interaction runs. The letter dominance threshold was set to 0.75 and the language dominance threshold was set to 0.5. Mean frequency and variance is the average and variance of the proportions of the dominating language.

inated, `c` letter dominated, `d` letter dominated, and `e` letter dominated. All languages that were not letter dominated were grouped into a single category called "other". This distribution will be called the language distribution for the agents.

A run of the simulation is *x language dominated* if the proportion of agents out of the total agents that are `x` letter dominated is greater than the *language dominance threshold.* If no letter `x` can be found that satisfies the language domination threshold, we say the run is not language dominated. If the language dominance threshold is set greater than 0.5, there will either be one unique letter `x` that will language dominate, or else the agent's language will not be dominated.

For each run of the simulation, we found the dominant language, or whether the run was not dominated.

Table 2 shows the distribution of dominating languages. In nearly all the frequency-dependent interaction runs, there was a dominating language. The average of the proportion of the dominating language is .7715. Breaking the runs down into the two clusters shown in Figure 2, we can see that in both clusters all the runs had a dominating language.

The one run that did not converge to a dominating run is the high outlier in Figure 2. In this case, the language distribution was split between an `a` language and a `b` language, which is why there is a high error for this run. This run resulted in a network with low betweenness centrality, just like the low outlier, but the difference is that in this run, the two most connected nodes in the network correspond to agents with different languages. In the low outlier, the most connected nodes all correspond to agents with the same language.

We also see that the proportion of agents speaking the dominant language is lower for cluster A than for cluster B. This may also be related to the slightly higher betweenness centrality of cluster A.

5. DISCUSSION AND FUTURE WORK

We have investigated a method of creating *emergent* topologies for agent interaction, and examined its effects on language convergence. We showed that by incorporating a frequency-dependent measure in the definition of fitness for an agent, we can generate scale-free topologies, which lead to quick convergence. This definition of fitness makes intuitive sense: if many agents are talking to a particular agent, then it pays to talk to that agent more, since it probably speaks a widely-spoken and useful language.

We saw however that emergent topologies can have varying degrees of success. In one case at least, the population converged to two distinct languages (the high outlier), because two highly-connected agents speak different languages. On the other hand, when multiple highly connected agents speak the same language, we saw that the population error drops to its lowest value (the low outlier). In general, the population converges quickly if there is a single dominant node in the population, which is what is observed in most simulation runs.

However, the tradeoff in this case is convergence vs robustness. Networks with high betweenness centrality, as are generally observed in our simulations, are very susceptible to attack, because removing the dominant node will cause the network to fall apart. This suggests one direction for future work: how do we converge to a stable language if agents have finite lifetimes?

The other observation is that the population converges to a "simple" language, i.e. a language consisting entirely of strings of a single symbol. This is not surprising since, in our simple experiments, most languages were of this kind. Random initializations of simple recurrent networks rarely result in more complex languages. Further, it may be the case that in the absence of some external pressure towards a complex language, populations always tend towards a maximally simple common language as it is the most easily learnable. This suggest another natural followup to the work described in this paper: how do we converge to a "good" language? The "goodness" of a language can be incorporated into our framework using the frequency-independent part of fitness for each agent. Thus, agents that are particularly good at solving some task and talking about it may have high frequency-independent fitness. Therefore we intend to study convergence in a situated population of agents, where the environment would provide the pressure towards a more complex language.

6. ACKNOWLEDGEMENTS

We thank the UIUC Language Evolution Group for very helpful and stimulating discussions. This work was supported in part by NSF grant IIS-0340996.

7. REFERENCES

[1] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, October 1999.

[2] M. Bodén and J. Wiles. Context-free and

context-sensitive dynamics in recurrent neural networks. *Connection Science*, 12(3,4):197–210, 2001.

[3] R. Bürger. Mathematical principles of mutation - selection models. *Genetica*, 102-103(0):279–298, January 1998.

[4] F. Cucker, S. Smale, and D.-X. Zhou. *Language Acquisition, Change and Emergence*, chapter Modelling Language Evolution. 2003.

[5] J. Delgado. Emergence of social conventions in complex networks. *Artificial Intelligence*, 141:171–185, 2002.

[6] J. L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[7] N. Komarova, P. Niyogi, and M. Nowak. Evolutionary dynamics of grammar acquisition. *Journal of Theoretical Biology*, 209(1):43–59, 2002.

[8] Y. Lee, T. C. Collier, C. E. Taylor, and E. E. Stabler. The role of population structure in language evolution. In *Proceedings of the 10th International Symposium on Artificial Life and Robotics*, 2005.

[9] K. M. Page and M. A. Nowak. Unifying evolutionary dynamics. *Journal of theoretical biology*, 219:93–98, 2002.

[10] Y. Shoham and M. Tennenholtz. On the emergence of social conventions: modeling, analysis, and simulations. *Artificial Intelligence*, 94(1-2):139–166, jul 1997.

[11] S. Swarup and L. Gasser. Unifying network and evolutionary dynamics. *Submitted*, 2005.