SIMPLE, BUT NOT TOO SIMPLE: LEARNABILITY *VS.* FUNCTIONALITY IN LANGUAGE EVOLUTION

SAMARTH SWARUP

Department of Computer Science,
University of Illinois at Urbana-Champaign,
Urbana-Champaign, IL 61801, USA
swarup@uiuc.edu

LES GASSER

Graduate School of Library and Information Science, and
Department of Computer Science,
University of Illinois at Urbana Champaign,
Urbana-Champaign, IL, 61801, USA
gasser@uiuc.edu

We show that artificial language evolution involves the interplay of two opposing forces: *pressure toward simple representations* imposed by the dynamics of collective learning, and *pressure towards complex representations* imposed by requirements of agents' tasks. The push-pull of these two forces results in the emergence of a language that is balanced: "simple but not too simple." We introduce the *classification game* to study the emergence of these balanced languages and their properties. Our agents use artificial neural networks to learn how to solve tasks, and a simple counting algorithm to simultaneously learn a language as a form-meaning mapping. We show that task-language coupling drives the simplicity-complexity balance, and that both compositional and holistic languages can emerge.

1. Introduction

In recent years, the application of the evolutionary metaphor to language change has gained currency. A natural question this raises is, what determines fitness of a language? Linguists often answer by attributing extrinsic sources of fitness, such as the prestige of the speaker (Croft, 2001; Mufwene, 2002).

For human languages it is generally accepted that intrinsic factors such as the learnability of a language do not vary across languages. A child born into a Hindi-speaking community will learn Hindi as easily as a child born into an English-speaking community will learn English. Modern languages, however, have adapted over a long period of time. If we go far enough into the history of language, it is clear that (aspects of) early languages had differential fitness. For example, Phoenicians were the first to develop a phonetic alphabet. This innovation quickly became established in (nearly) all languages, even though the Phoenician language itself died out. One explanation is that phonetic alphabets fixated because a phonetic writing system is much easier to learn.

However, if learnability were the *only* source of fitness for a language, we would expect to see maximally simple, possibly trivial, languages prevail, since these can be learned most easily. Indeed, simulations have shown this to be the case (Swarup & Gasser, 2006). To allow the emergence of more complex, and thus more useful, languages, a bias for increasing complexity has to be built in by the experimenter (Briscoe, 2003). Where would this bias come from, if it exists, in an evolutionary system?

It seems intuitive that the counter-pressure to make language more complex must come from the functionality of language. A language is *for* something. In other words, the agents gain some benefit from having a particular language/representation. If the use of a particular language gives an agent high reward (perhaps through low error on some task), then part of that reward gets transferred to the language as a boost in fitness. Languages that are too simple, however, are unlikely to be very functional, because their information-carrying capacity is low; an agent should feel a pressure to discard such a language.

Thus we can imagine that languages occupy a complexity line, with complexity increasing to the right, as shown in figure 1. Learnability increases with simplicity, to the left, and expressiveness or functionality increases with complexity, to the right. Together, these two define the intrinsic fitness of languages.

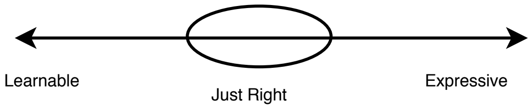

Figure 1. The complexity line for languages.

In terms of this complexity line, we would like the languages that evolve to be in the region that is "just right", where the language that evolves is both easily learnable and adequately useful. Such a language would be well-adapted, by some measure, to the tasks at hand.

The goal of this paper is to relate language to task in a way that allows a population of agents to jointly learn a shared representation that is well-adapted to the complexity of the task they face. We do this by setting up a *classification game* (below), where agents interact with each other while learning to perform some classification task. The interaction between agents results in the emergence of a shared representation, or language, that is simple but not too simple.

The rest of this paper is organized as follows. First we relate language to task by treating the agents' hypothesis space as an information channel. We present experiments that illustrate the kinds of languages that can emerge with and without interaction between agents. We show that both holistic and compositional languages can emerge, and that the emergent languages are more efficient than representations learned without communication. We summarize contributions, and speculate on future work.

2. Framing Learning in Terms of Communication

Our agents learn to perform classification tasks. They search through a hypothesis space, $\mathcal{H}$, to find the best classifier for a given (training) data set of n examples, and using it to label the data. We relate a language to this task by treating $\mathcal{H}$ as an information channel. In this communication-oriented view of learning, an agent, "Alice," sends her labels for all the examples to another agent, Bob. Alice can, instead of transmitting the labels directly, send the appropriate hypothesis instead, which allows Bob to reconstruct the labels himself. The advantage of this is that it allows Bob to generalize to unseen examples. It can be shown that the "simpler" the hypothesis that Alice sends, the better Bob will be able to generalize.

So, if we think of the hypothesis class as a channel, its channel capacity is the logarithm of a quantity known as the *growth function*, $S_{\mathcal{H}}(n)$, of the hypothesis class. For n points, $S_{\mathcal{H}}(n)$ is the number of ways in which hypotheses from $\mathcal{H}$ can classify the points into two classes. When the growth function is less than 2^n, some of the dichotomies cannot be captured by hypotheses in $\mathcal{H}$. In a sense, the channel capacity is not high enough. However, in this case the learner can use a longer channel code, thereby reducing the bit rate, to transfer any of the dichotomies. This means that the learner would only transfer some of the labels, or possibly *part of a label*, at each step. Using a longer channel code corresponds to using multiple hypotheses.

3. The Classification Game

Now we describe the experimental setup in which agents interact with each other and learn to solve a classification problem while also learning to communicate about it. The learning task in all the experiments below (except the first), is the XOR classification task; it is well known in classification learning and its results are easy to visualize. Inputs consist of just two bits. Thus there are four possible inputs: 00, 01, 10, and 11. The output, or label, is 1 if the inputs are different, otherwise it is zero.

We choose hyperplanes (i.e. straight lines) as our hypothesis class. One crucial consequence of this is that at least two hypotheses are needed to solve the XOR task, as will be obvious from the figures later. This choice of hypothesis space leads very naturally to an artificial neural network (ANN) implementation. Each hidden layer node of an ANN, called a *perceptron*, can be thought of as representing a hyperplane. The number of hidden layer nodes in each agents' neural network defines the maximum number of hyperplanes it can use to classify the training examples. We also refer to the hidden layer as the *encoder*, since it encodes the inputs into features that are to be communicated.. The second, or output layer, has just a single node. We also refer to this as the *decoder*, since it decodes the features extracted by the encoder to find the output. The agents also convert the outputs of the encoder layer into a public language using a learned form-meaning mapping (FMM), a matrix $[f_{ij}]$ where each entry defines the likelihood of pairing that form (typically a letter of the alphabet), with that meaning (hidden layer node number).

The game protocol is simple. At each step, we select two agents uniformly randomly. We assign one agent to the role of speaker, and the other to hearer. Next, we present both with the same training example. The speaker feeds its encoder outputs through its form-meaning map to generate an *utterance* in (its version of) the emerging public language. The hearer tries to decode this utterance via its own form-meaning mapping, and uses its decoder to generate a label. We then give both agents the expected label; they calculate error and update their neural networks and form-meaning mappings. Figure 2 shows the process.

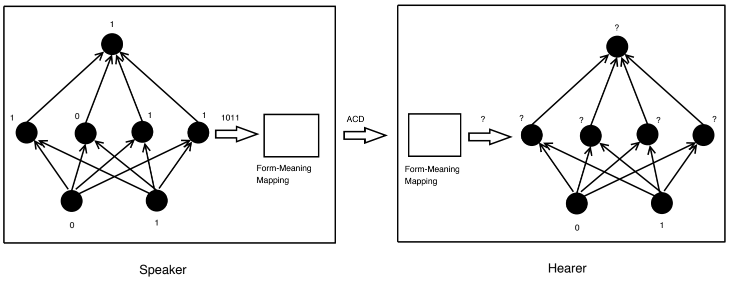

Figure 2. Speaker-hearer interaction.

The form-meaning mapping is updated with a counting algorithm. Whenever a particular form and meaning pairing is observed, an agent adds a constant δ to that particular FMM locus. At the same time, all the other entries in the same row and the same column are decremented by a smaller constant, ϵ. This discourages synonymy and polysemy, and is inspired by a mutual exclusivity bias seen in the language acquisition behavior of young children (Markman & Wachtel, 1988). Agents update their form-meaning maps while speaking and while hearing. Further details can be found in (Swarup, 2007).

4. Experiments

Now we show the results of a series of experiments which demonstrate the effects of collective learning on the emergent representations.

4.1. *Driving simplicity: communication without task learning*

This experiment shows how collective language learning in the ungrounded case (without environmental feedback) leads to linguistic simplicity. At each step, we give the speaker a random 2-bit Boolean vector, and the label generated by the speaker is treated as the expected label for the hearer's neural network. Thus the hearer tries to update its neural network to match the speaker's. Speaker and hearer both update their form-meaning mappings as described previously. An example representation that the agents converge upon is shown in figure 3.

Input	Label	Utterance
(0, 0)	0	C
(0, 1)	0	C
(1, 0)	0	C
(1, 1)	0	C

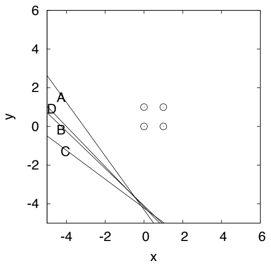

Figure 3. The result of ungrounded language learning. The agents come to consensus on a trivially simple mapping, which assigns the same utterance, and same label, to all the points.

We see that all the hyperplanes have been pushed off to one side. It so happens that hyperplanes A, B, and D are oriented in such a way that they label all the points as zero, while hyperplane C has ended up in the reverse orientation with respect to the others, and labels all the points as one. This is why the table shows C as the utterance for all the points. The decoder for all the agents, though, decodes this as the label zero for each point, as shown in the table, and also by the empty (unfilled) circles in the figure.

4.2. *Driving complexity: task learning without communication*

The second experiment shows the opposite: pressure towards expressiveness. In this case agents all learn individually to solve the XOR problem from training examples, and they don't communicate at all. Figure 4 shows an example of an individually learned solution to the task. They are not updating their form-meaning mappings and thus it does not really make sense to talk about their languages in this case. But, if we assume a mapping that assigns symbol A to the first hidden layer node, B to the second and so on, we can derive what their language would have been if this solution had been adopted by all agents.

The agents had four hidden layer nodes available to them to encode hypotheses and this agent uses them all. While it solves the problem perfectly, the learned

Input	Label	Utterance
(0, 0)	0	ABC
(0, 1)	1	C
(1, 0)	1	BD
(1, 1)	0	B

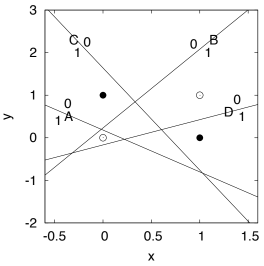

Figure 4. A learned solution to the XOR problem, without communication between agents. Different agents learn different solutions, but they all generally learn overly complex solutions.

representation, as we can see, is overly complex. Different agents learn different solutions, depending on random initialization of ANN weights. However, the minimal solution, which uses only two hyperplanes, is observed very rarely.

4.3. *Finding balance: coupled task-communication learning*

In the next experiment, we allow the agents to communicate while *also* learning to solve the task. With this task-language coupling, the agents converge to a maximally simple mapping that also solves the problem. In some (chance-determined) runs, agents develop languages with redundant symbols, and in some they do not; a redundancy example is shown in figure 5.

Input	Label	Utterance
(0, 0)	0	ABCD
(0, 1)	1	ABD
(1, 0)	1	ABD
(1, 1)	0	BD

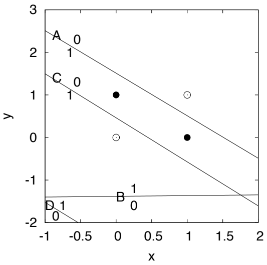

Figure 5. A learned solution to the XOR problem, with communication between agents. All agents converge to the same solution. Even though they have four hidden layers nodes, they converge on a simpler solution that uses only two of the nodes.

The population consisted of only four agents, and figure 6 shows the learned form-meaning mappings of all the agents, as Hinton diagrams. The size of a box is proportional to the magnitude of the value.

There are a couple of interesting things to note about these matrices. First, they all map symbols and hyperplanes uniquely to each other. Each row and col-

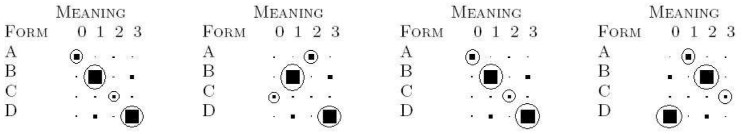

Figure 6. The learned form-meaning matrices for each of the agents from experiment 4. Form-meaning pairs that have become associated have been highlighted with circles.

umn has a distinct maximum in each of the matrices. Second, they are all different (except the first and third). In other words, their private interpretation of symbols is different, even though they all understand each other and have perfect performance on the task. Thus while their task representations and public language are aligned, their private languages are different.

4.4. *Coupled learning: the emergence of a holistic language*

The language that is shown to emerge in the previous experiment is compositional, in a sense. The agents make use of multiple symbols, and they combine them meaningfully to communicate about the labels of various points. Though this is an interesting and desirable outcome from the point of view of language evolution, it is a pertinent question to ask whether this outcome is in some way built in, or whether it is truly emergent.

To show that it is, in fact, emergent, we present the following result. Figure 7 shows the outcome of a run with identical parameters as experiment 4. However, this time we see the emergence of a holistic language. Each point has a unique symbol associated with it (one of the points has no symbol, as A is redundant).

Input	Label	Utterance
(0, 0)	0	AC
(0, 1)	1	AB
(1, 0)	1	AD
(1, 1)	0	A

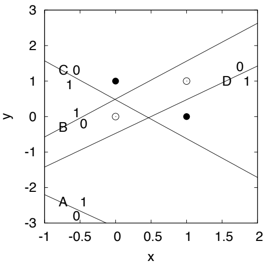

Figure 7. A learned solution to the XOR problem, where the communication matrix is learned by counting. All agents converge to the same solution. They essentially memorize the points, assigning one hidden layer node to each point.

In effect, the agents have memorized the points. This is only possible in the case where the neural network is large enough in the sense of having enough hidden layer nodes to assign a unique one to each point. In any realistic problem, this is generally not the case. However, this notion of the role of cognitive capacity in the emergence of compositionality and syntax has been studied theoretically by Nowak et al. (Nowak, Plotkin, & Jansen, 2000). They showed that when the number of words that agents must remember exceeds a threshold, the emergence of syntax is triggered. In our case, this threshold is defined by the number of hidden layer nodes. If the number of points that must be labeled exceeds the number of hidden layer nodes, the network must clearly resort to a compositional code to solve the task.

5. Conclusion

We have presented a principled way of connecting language to task. We have demonstrated that this results in a complexity regularization effect. The representations that emerge are close to optimal for the given task. Seen as languages, they can evolve to be either holistic or compositional, depending on initial conditions and cognitive capacity. More experiments, and more discussion, can be found in (Swarup, 2007), including a demonstration of the emergence of rudimentary grammar, which emerges, surprisingly, without having to include any syntactic aspect to the linguistic processing done by the agents. Future work involves developing an information-theoretic account of the effects observed, and an analysis of the dynamics of language learning, which have been observed to exhibit some interesting oscillatory phenomena.

References

Briscoe, T. (2003). Grammatical assimilation. In M. H. Christiansen & S. Kirby (Eds.), *Language evolution: The states of the art.* Oxford University Press.

Croft, W. (2001). *Explaining language change.* Longman Group United Kingdom.

Markman, E. M., & Wachtel, G. F. (1988). Children's use of mutual exclusivity to constrain the meanings of words. *Cognitive Psychology*, *20*, 121-157.

Mufwene, S. (2002). Competition and selection in language evolution. *Selection*, *3*(1), 45-56.

Nowak, M. A., Plotkin, J. B., & Jansen, V. A. A. (2000). The evolution of syntactic commmunication. *Nature*, *404*, 495-498.

Swarup, S. (2007). *Artificial language evolution on a dynamical interaction network.* Unpublished doctoral dissertation, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA.

Swarup, S., & Gasser, L. (2006). Noisy preferential attachment and language evolution. In *From animals to animats 9: Proceedings of the ninth international conference on the simulation of adaptive behavior.* Rome, Italy.