# Evolution of Syntax Through Horizontal Social Interactions

Arjun Karande
Kunal Kala

April 24, 2005

## 1 Introduction

Human language is a unique communication system[1]. It has got a syntactical structure with properties like compositionality and recursion. This makes possible the synthesis of an infinite range of expression, which is unique to humans. Humans have the power to learn the signal-meaning mapping through the observation of other's use of language which also is absent in other animals. The interactions between these two have been long studied and the emergence of syntax has often been attributed to the complex dynamical learning process found in the humans [1, 2, 3]. This approach proposes that syntax has evolved without natural selection i.e. without the development of an innate language acquisition device (LAD).

The dynamical process of language transmission has been modeled in different social contexts. In [1], the transmission is vertical i.e. from one individual in a generation to a blank learner in the next generation.

In [2], Kirby and Simon have tried to model horizontal transmission in same generation of learners. But, in this case the transmission has direction. Any blank individual entering the pool can learn only from its immediate neighbours. We argue that such a model is similar to the vertical model. Now the learner has access to two individuals instead of one. This cannot be called perfectly horizontal.

In our project we have tried to model perfectly horizontal transmission in which a blank learner entering the population can access any individual for

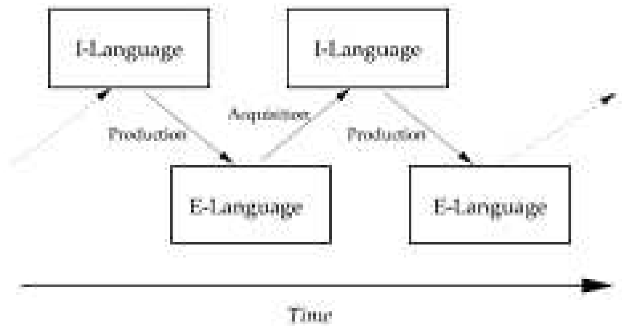



Figure 1: Transmission of language over time

learning. We want to show that compositionality properties of syntax will inevitably emerge over time in this case also.

When a blank learner has access to only one individual [1] or two individuals [2], poverty of stimulus plays an important role in the language transmission process. We wish to see the extent of influence of poverty of stimulus and role of subsumption rules in the emergence of syntax in our model.

# 2 Computational Model

Our model works within the framework shown in the following figure.

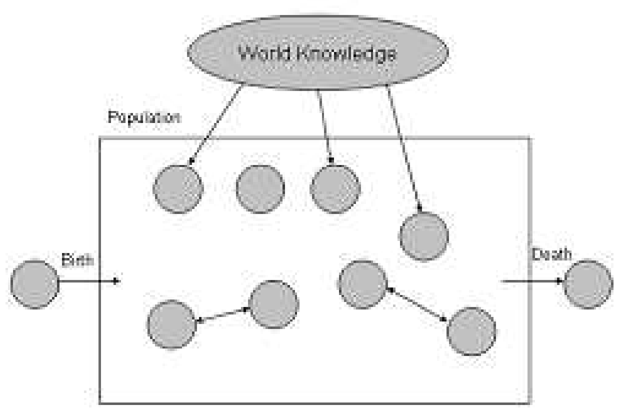



Figure 2: Computation Model I

The population consists of individuals called agents. Blank individuals can enter the population and individuals can also leave. There is a world knowledge which is the set of all possible meanings. Subsets of this set are possessed by agents. There are n iterations in which two agents are picked at random and made to interact m times. The interaction between two agents can be modeled as the following diagram.

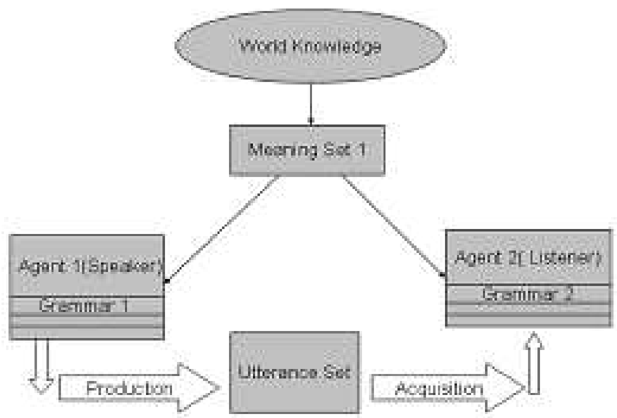



Figure 3: Computation Model I

Our computational model to imitate interaction between two agents is very similar to the one in [1]. The simulation implements the following processes:

1. For a particular interaction, the shared knowledge is the meaning set in context. A set of meaning from this shared knowledge is chosen randomly and given to speaker to express.

2. The speaker then attempts to express each meaning either using its own internalized knowledge of language or by some random process of invention.

3. Now the listener gets this utterance(meaning-signal pair) and tries to find an already internalized rule which could have been responsible for this mapping. If a rule is not found the meaning-signal pair is induced.

In our simulations the world is made of predefined atomic concepts like:

john, tiger, eats, fear

These concepts are combined into predicate-argument combinations, which may have hierarchical structure. For example:

$fears(john, tiger)$
$knows(john, eats(tiger, john))$

An utterance is a meaning-signal pair. For example:

$< johneatsmary, eats(john, mary) >$

# 3 Learning

The learning algorithm may be discussed in the following contexts:

## 3.1 Grammar Representation

The rules internalized are context free grammar, to be more specific they are definitive clause grammar (DCG) in which semantic arguments are attached to the non terminals. They could be either holistic or compositional in nature. For example:

1.$S/eats(tiger, john)- > tigereatsjohn$
(holistic)

2.$S/p(x, y)- > N/xV/pN/y$
$V/eats- > eats$
$N/tiger- > tiger$
$N/john- > john$
(compositional)

## 3.2 Rule Subsumption

Initially the grammar of every individual has no rules and utterances induce rules trivially. For example, the utterance:
$< tigereatsmary, eats(tigers, mary) >$

is induced as:

$S/eats(tigers, mary) -> tigereatsmary$

But, with such rules the internalized grammar will bloat up. So the learner must have the power to generalize. This is done by the subsumption rule, whereby new rules are incorporated, general rules are found which subsume two or more rules and then duplicated rules are deleted. For example the following set of rules:

$S/eats(tiger, sausages) -> tigeeatssausages$
$S/eats(john, sausages) -> johneatssausages$

can be replaced by:

$S/eats(x, sausages) -> N/xeatssausages$
$N/tiger -> tiger$
$N/john -> john$

Similarly, subsumption may be applied to a single rule. In many cases, a simple generatlization might be applied by incorporating one rule within another. For eg. given the following rules:

$S/eats(tiger, sausages) -> tigereatssausages$
$N/tiger -> tiger$

a learner might want to combine these two into:

$S/eats(x, sausages) -> N/xeatssausages$
$N/tiger -> tiger$

Another subsumption rule is the merging rule. Here rules differing only in the non terminal are subsumed by a general rule. For example:

$N/mary -> mary$
$M/mary -> mary$

can be replaced by single rule with N or M as the non terminal.

There can be many more subsumption rules apart from the three described above. In our simulations we have restricted ourselves to these three rules, as described in [1].

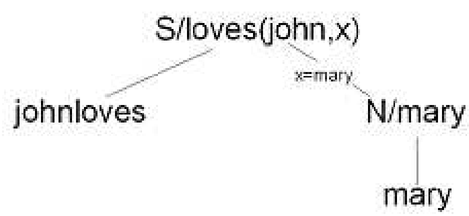
### 3.3 Invention

If the speaker does not have a way to produce a string given a meaning i.e. the required rule is absent in the grammar, he or she finds the closest meaning for which a rule is available. A parse tree for the meaning is then created. At the wrong part in the tree the string is replaced by a random sequence.

Let us look at an example from [1]. A speaker is asked to produce a string for the meaning $loves(john, anna)$. Suppose the speaker doesn't have the required grammar but he has the following rules:

$S/loves(john, x) -> johnlovesN/x$
$N/mary -> mary$

Thus the nearest meaning for which the speaker can produce a string is $loves(john, mary)$. A parse tree is created for this meaning:




The wrong part in this parse tree is the node where mary is there. So this can be replaced by a random sequence of characters. So the invented string for $loves(john, anna)$ might be *johnlovesrtui*. Finally, the invented rule itself is induced in the grammar.

## 4 Summary of the Simulation Cycle

The simulation goes through the following steps:

1. Initialize a population with no internal language.

2. Repeat n times:

3. Pick 2 agents randomly from the population. One speaker, other listener

4. Perform m interactions.

5. Kill a random agent with some probability

# 5 Results

The results shown here are for two experiments that were carried out. To show the trend in the grammer learnt over generations, we plotted the average grammar size against the iteration number.

## 5.1 Experiment 1

Firstly, we tried the model described above without killing off individuals. The number of individuals is 10, the number of interactions between a pair of individuals is 50, and the number of such iterations is 1000. The results show an explosion in the grammar size, with no sign of stability. The plot here is for the first 100 iterations only, but a similar trend is seen even beyond this point. The initial stable grammar obtained in [1] was nowhere to be seen:

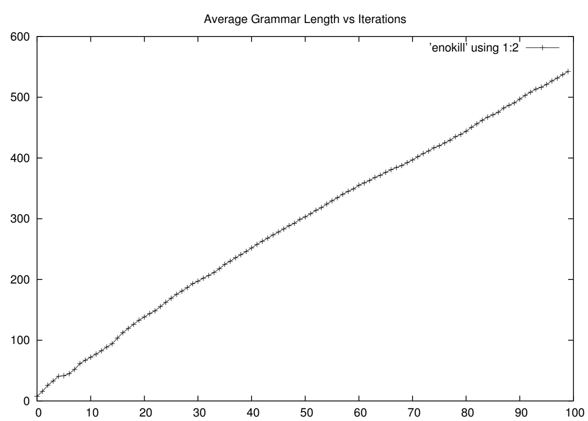



Figure 4: Results for experiment 1

## 5.2 Experiment 2

We next varied the probability of killing off individuals. The probability was set to 0.3 first, keeping the other parameters constant. The results shown indicate some emergence of syntax, but closer examination of the inner grammatical rules show that although there are rules that result from the subsumption rules, a stable grammatical system is not reached. The plot of average grammar size vs. iterations is presented:

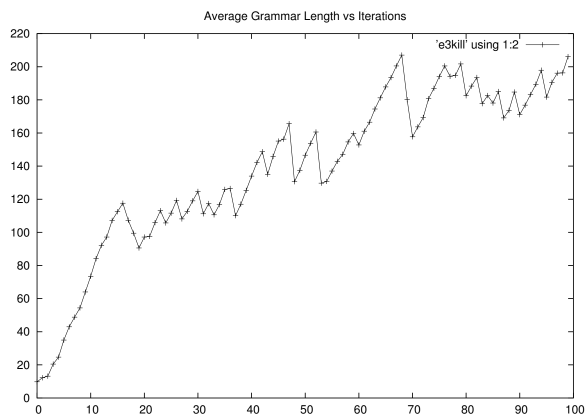



Figure 5: Results for experiment 2

A sample of the grammar emerging from 100 iterations is:

$S/loves(bob, alice)- > gqjnj$<br>
$G/alice- > kn$<br>
$G/mary- > tl$<br>
$S/eats(x, parker)- > oG/xn$<br>
$F/mary- > qoloh$<br>
$S/eats(x, bob)- > F/xl$<br>
$E/mary- > dcetl$<br>
$E/alice- > sftdps$<br>
$S/eats(bob, y)- > E/yg$

## 5.3 Experiment 3

To show how the probability of killing, which we believe is related to lack of stimulus and forms a bottleneck in learning, affects the learning, we varied the probability to 0.6. We obtained the following trend:

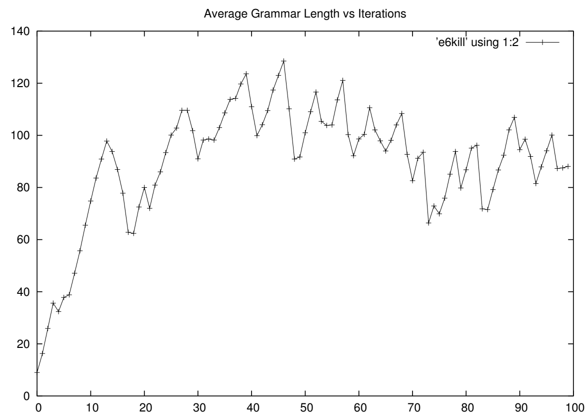



Figure 6: Results for experiment 3

Again, this graph, although showing a slower rate of growth and seemingly stable grammar size, does not stabilize even into an initial grammar. Again, a sample of the grammar is listed below. Although a formal measurement of commonality between the grammars (i.e. by calculating the intersection of the language produced by the grammar) was not carried out, the following rules were seen to be repeated in a particular run:

$T/alice- > ghb$
$T/mary- > qdhn$
$S/hates(bob, y)- > aT/yi$
$S/alice- > bgga$
$S/mary- > q$
$S/loves(john, y)- > aS/yq$

There was no formation of rules with the semantics $p(x, y)$, although rules containing semantics of the form $eats(x, y)$ were seen. Few non-terminals showed unification, but an overall trend wasn't seen.

# 6 Conclusion

Kirby's models[1, 2, 3] are intuitive and seemingly straight-forward. It was our notion that the emphasis Kirby placed on the role of bottlenecks was not necessary, and wished to prove that horizontal interactions driven by the subsumption assumption would suffice to evolve syntax. However, results shown by our experiments were largely disappointing, showing the formation of certain rules that are relatively stable and used often, but lacking an overall structure.

Further, the effect of allowing the system to be an open system, i.e. letting new individuals come in and old ones die out was not very profound. Increasing the rate to 0.6, which corresponds to killing an individual 6 times out of every 10, reduced the size of the grammar, but did not affect its growth substantially.

Although syntax did not emerge from pure horizontal interactions, we do see the intial traces of the formation of a common grammar set, and perhaps with vaster subsumption rules (perhaps general unification) and longer iterations, a stable pattern may be achievable, although there is no sure indication that this will happen.

# References


[1] Simon Kirby. Learning, bottlenecks and the evolution of recursive syntax. In *Linguistic Evolution through Language Acquisition: Formal and Computational Models*, 1999.

[2] Simon Kirby. How compositionality emerges from vocabulary in a population of learners. In *The Evolutionary Emergence of Language: Social Function and the origins of linguistic form*, 2000.

[3] Kenny Smith, Simon Kirby, and Henry Brighton. Iterated learning: A framework for the emergence of language. In *Artificial Life*, 2000.