# "The limits of my world are the limits of my language" – An agent based exploration of the Naming Game in a world of complex context*

Peter Graff
Department of Linguistics and Philosophy
Massachusetts Institute of Technology
graff@mit.edu

Kathleen Sprouffske
Genomics and Computational Biology Program
University of Pennsylvania
sprouffske@gmail.com

François Ascani
Department of Oceanography
University of Hawai'i
fascani@hawaii.edu

Petr Švarc
Institute of Economic Studies
Charles University Prague
petrsvarc@email.cz

September 14, 2008

## 1 Introduction

Language is a cultural phenomenon arising through extensive abstraction and generalization in a cooperative species. The goal of this study is to test the minimal mathematical prerequisites for the emergence of language-like communication in a virtual population of agents. With a minimum amount of biases, we test the effects of the conceptualization of the world on language emergence and come to the conclusion that dependency in context could be the reason for grammatical dependency. All simulations were implemented in NetLogo 4.0.2.

## 2 The Naming Game

The Naming Game (Kaplan et al., 1998) is an exploration of one possible scenario for language emergence. There are several version of the Naming Game and the algorithm described below is the particular version we chose as the basis for our investigation. The idea is to construct and artificial populations of agents which have a limited number of concepts stored in their mental lexicon. Each concept is associated with a distribution of symbols. Agents are paired and have "conversations" about a randomly chosen context. The agent assigned the role of speaker randomly chooses a symbol from the distribution of symbols associated with a concept in the context, which the agent assigned hearer adds to his distribution of the same concept. As agents interact more and more they converge on a symbol for each concept. Throughout the paper "convergence" of a simulation refers to a state in which all agents have settled on a common expression for each concept. The detailed algorithm is given in (1).

(1)
a. Create a world of concepts $C_1$ to $C_n$.
b. Create Agent population with grammar $G$ consisting of symbol vectors $c_1$ to $c_n$ associated with a concept each.
c. Randomly select two agents. Assign one agent speaker $S$ and the other hearer $H$.
d. Randomly assign a concept $C_x$ context.
e. Choose a symbol from distribution $c_x$ of $S$.
f. Add said symbol to distribution $c_x$ of $H$.
g. Delete oldest symbol in distribution $c_x$ of $H$.
h. Repeat (2-c) to (2-g) until convergence.

Convergence is always guaranteed as the Naming Game mathematically draws on Pólya's urn, which

*This work was partially supported by the Santa Fe Institute whose research and education programs are supported by core funding from the National Science Foundation and by gifts and grants from individuals, corporations, other foundations, and members of the Institute's Business Network for Complex Systems Research.

has been proven to converge (Gouet, 1997). In this thought experiment an urn originally contains a red and a blue ball. An algorithm picks one ball at random and returns the ball with another ball of the same color to the urn. The probability of one color will exceed the probability of the other after a certain amount of time. If older balls are removed, as is the case in our particular implementation, convergence on a single color is guaranteed after a certain amount of time.

The Naming Game provides a good basis for the investigation of language emergence. It, however, omits a crucial property of language and the world, the ambiguity of context. The next section describes the results we obtained simulation the Naming Game in a complex world.

# 3 The Naming Game in a Complex World

In this simulation we modified the world in which the Naming Game takes place. Instead of having several discrete contexts and unambiguous concepts we introduced two types of concepts, objects and properties which could co-occur. Each context then contained an object and a property. To give agents a chance to produce both meanings in a single utterance we increased the number of symbols uttered in each conversation from 1 to 2. The modified algorithm is given in (2).

(2)
a. Create a world of objects $O_1$ to $O_n$ and properties $P_1$ to $P_n$.
b. Create Agent population with grammar $G$ consisting of symbol vectors $o_1$ to $o_n$ associated with an object each and symbol vectors $p_1$ to $p_n$ associated with an property each.
c. Randomly select two agents. Assign one agent speaker $S$ and the other hearer $H$.
d. Randomly assign an object-property pair $< O_x, P_y >$ context.
e. Choose two symbols from the union of the distributions $o_x$ and $p_y$ of $S$.
f. Add said symbols to distributions $o_x$ and $p_y$ of $H$.
g. Delete oldest symbols in distributions $o_x$ and $p_y$ of $H$.
h. Repeat (2-c) to (2-g) until convergence.

This particular implementation converges on total homonymy. Since the concepts overlap and all distributions are united with each other at some point all distributions will converge on a single symbol. The next section describes a model that achieves non-unique convergence in a world of overlapping concepts.

# 4 Model A: Comprehension Bias

In this model we added a comprehension bias to the symbol addition step in the algorithm. The rationale behind the bias is as follows; agents assume that other agents have similar distributions as they do and try to reinforce these distributions. The bias is given in (3).

(3) **Comprehension Bias**
Add a symbol to the particular distribution where it is represented the most.

Consider the following example. An agent hears utterance = [ 2 5 ] in the context $< O_x, P_y >$. The agents grammar contains $o_x$ = [ 2 2 2 1 3 ] and $p_y$ = [ 5 5 2 2 1 ]. Under the assumption of the comprehension bias the agent assumes that the speaker has identical distributions to himself and tries to enforce maxima in these distributions. The hearer thus checks for each symbol in the utterance to which distribution he should add it by comparing the frequency of each symbol with the frequency of that symbol in the two distributions referenced by the context. In our example [ 2 ] is more frequent in the hearer's vector $o_x$ and [ 5 ] is more frequent in the hearer's vector $p_y$. Therefore the hearer adds [ 2 ] to $o_x$ resulting in [ 2 2 2 2 1 ] after deletion of the oldest symbol and [ 5 ] to $p_y$ resulting in [ 5 5 5 2 2 ] after deletion of the oldest symbol. If the frequencies are equal the agent randomly chooses the distribution to which he adds the number. The modified algorithm is given in (4).

(4)
a. Create a world of objects $O_1$ to $O_n$ and properties $P_1$ to $P_n$.
b. Create Agent population with grammar $G$ consisting of symbol vectors $o_1$ to $o_n$ associated with an object each and symbol vectors $p_1$ to $p_n$ associated with an property each.
c. Randomly select two agents. Assign one agent speaker $S$ and the other hearer $H$.

d. Randomly assign an object-property pair $< O_x, P_y >$ context.
e. Choose two symbols from the union of the distributions $o_x$ and $p_y$ of $S$.
f. For each of said symbols compute whether said symbols occur more often in $o_x$ or $p_y$ of $H$.
g. Add each symbol to the distribution where it occurs more often. If a symbol occurs equally often in both distributions randomly chose a distribution and add it to that distribution.
h. Delete oldest symbols in distributions of $H$ to which symbols were added.
i. Repeat (4-c) to (4-h) until convergence.

It turns out that this simple addition to the algorithm ensures non-unique convergence in a complex world. In fact we found, that with this bias, adding complexity decreases convergence time. The more different object/property pairs can appear together as context, the faster agents settle on particular symbols to refer to them.

# 5 Model B: Conditional Probabilities

All models so far have had agents produce two unordered symbols at each interaction. This is not the case in natural language. Different sentences, phrases, words, syllables right down to the level of individual sounds are ordered and dependent on each other to express meaning. In order to simulate these dependencies we introduced conditional probabilities into our next model. In addition to the one main vector storing the symbol distribution for each concept, we have one conditional vector for each symbol. An example of the distributions for some concept in this model is given in (5).

(5) Set of symbols = { 1, 2, 3, 4 }
0: [ 2 2 2 2 2 ]
1: [ 3 1 4 3 4 ]
2: [ 1 1 1 1 1 ]
3: [ 1 4 4 3 4 ]
4: [ 1 2 4 2 4 ]

The primary vector labeled 0 in (5) is the distribution from which the first symbol is chosen. Depending on what the first symbol is chosen the agent chooses the second symbol from the distribution referenced by that symbol. An agent with the grammar in (5) would always produce the utterance [ 2 1 ] in the relevant context. Learning of conditional distributions takes place in the same way as before. The complete algorithm is given in (6).

(6)
a. Create a world of objects $O_1$ to $O_n$ and properties $P_1$ to $P_n$ and $s$ symbols.
b. Create Agent population with grammar $G$ consisting of a single primary vector $o_{x:0}$ and one conditional vector for each symbol $o_{x:1}$ to $o_{x:s}$ associated with an object each and a single primary vector $p_{y:0}$ and one conditional vector $p_{y:1}$ to $p_{y:s}$ for each symbol associated with a property each.
c. Randomly select two agents. Assign one agent speaker $S$ and the other hearer $H$.
d. Randomly assign an object-property pair $< O_x, P_y >$ context.
e. Choose a symbol from the union of the primary distributions $o_{x:0}$ and $p_{y:0}$ of $S$. Call it first symbol $m$.
f. Choose a symbol from the union of the distributions associated with the symbol chosen as first symbol $o_{x:m}$ and $p_{y:m}$ of $S$. Call it second symbol $l$.
g. For first symbol $m$ compute whether said symbol occurs more often in $o_{x:0}$ or $p_{y:0}$ of $H$.
h. For second symbol $l$ compute whether said symbol occurs more often in $o_{x:m}$ or $p_{y:m}$ of $H$
i. Add each symbol to the distribution where it occurs more often. If a symbol occurs equally often in both distributions randomly chose a distribution and add it to that distribution.
j. Delete oldest symbols in distributions of $H$ to which symbols were added.
k. Repeat (6-c) to (6-j) until convergence.

(7)

| | Object 1 | Property 1 |
|---|---|---|
| 0: | [ 2 2 2 2 2 ] | [ 3 3 3 3 3 ] |
| 1: | [ 3 1 4 3 4 ] | [ 4 4 4 4 4 ] |
| 2: | [ 1 1 1 1 1 ] | [ 1 1 1 1 1 ] |
| 3: | [ 2 2 2 2 2 ] | [ 2 2 2 2 2 ] |
| 4: | [ 4 4 4 4 4 ] | [ 1 2 4 2 4 ] |
| | Object 2 | Property 2 |
| 0: | [ 1 1 1 1 1 ] | [ 4 4 4 4 4 ] |
| 1: | [ 3 3 3 3 3 ] | [ 2 2 2 2 2 ] |
| 2: | [ 1 4 4 3 4 ] | [ 1 1 1 1 1 ] |
| 3: | [ 3 3 3 3 3 ] | [ 4 2 3 1 4 ] |
| 4: | [ 2 2 2 2 2 ] | [ 2 2 2 2 2 ] |

The result of this model is that agents converge on 4 different ways of expressing complex objects. Agents converge on some symbol for every primary vector and on some symbol for every vector referenced by some primary vector associated with the same concept or a concept that can co-occur with that concept. A sample grammar is given in (7). What is important to notice is that conditional vectors referenced by other members of the same class (e.g. other objects) do not converge. The resulting grammar has four possible outcomes for each concept as the unified distributions consist of symbols from both object and property distributions. In the case in (7) a complex context $< O_2, P_1 >$ could come out as [ 1 3 ], [ 3 2 ], [ 3 3 ] or [ 1 4 ] with 25% probability each. The result is not satisfying in that there is no consistent way to communicate a particular context.

# 6 Model C: Homonymy Bias

In the final model we added another bias to the conditional probabilities model. We term this bias "Homonymy Bias". This bias is given in (8).

(8) **Homonymy Speaking Bias**
Produce the particular symbol that is most unique.

Speakers do not chose according to frequency of symbols in the context distribution but chose the most unique symbol for each concept. The way this is done is by comparing across concepts.

(9) Homonymy Speaking Bias

a. Let $S_1(c_x)$ be the frequency of symbol 1 in distribution $c_x$.

b. Chose symbol $s_x$ in distribution $c_x$ of concept $X$ for which

$$\max(S_x(c_x) - \max_{i=0}^{n}(S_x(c_i)))$$

c. If several symbols fulfil this condition, choose the one that is more frequent in $c_x$.

d. If several symbols fulfil this condition, choose at random among them.

It turns out that this bias has an interesting effect on convergence in certain worlds. In all the simulations previously described, there were two distinct types of concepts, which could co-occur with concepts from the other class but not with concepts from their own. This means that a context could for example consist of an object and a property but not two properties or two objects. We, however, also ran simulations with different world structures. In one particular world objects could also occur by themselves. The convergence patter of simulations in this world, in conjunction with the homonymy bias and conditional probabilities produces something similar to grammatical dependency. Conditional symbols only converge when referenced by objects. Consider the following pattern in (10).

(10)

| | Object 1 | Property 1 |
|---|---|---|
| 0: | [ 2 2 2 2 2 ] | [ 3 3 3 3 3 ] |
| 1: | [ 3 1 4 3 4 ] | [ 4 4 4 4 4 ] |
| 2: | [ 2 2 2 2 2 ] | [ 1 1 1 1 1 ] |
| 3: | [ 1 2 3 2 1 ] | [ 4 3 2 1 1 ] |
| 4: | [ 4 1 3 2 4 ] | [ 1 2 4 2 4 ] |
| | Object 2 | Property 2 |
| 0: | [ 1 1 1 1 1 ] | [ 4 4 4 4 4 ] |
| 1: | [ 3 3 3 3 3 ] | [ 2 2 2 2 2 ] |
| 2: | [ 1 4 4 3 4 ] | [ 1 1 1 1 1 ] |
| 3: | [ 4 3 4 2 2 ] | [ 4 2 3 1 4 ] |
| 4: | [ 1 2 1 1 4 ] | [ 3 2 1 1 2 ] |

In a complex context such as for example $< O_1, P_1 >$ there are now only two converged utterances available, [ 2 2 ] and [ 2 1 ]. If the agent chooses to start the utterance with the primary symbol of the property the second symbol cannot be chosen consistently. This means that under selective pressure starting with the property symbol might be selected against. This is also the case in natural language where phrase heads are usually found consistently on one side of the phrase. Chomsky and Lasnik (1993) refer to this as the head parameter which decides whether phrase heads, such as the noun, in a noun-adjective phrase, appear to the left (as in Italian *cosa*

*bella*, "beautiful thing") or to the right (as in German *schöne Sache*, "beautiful thing") of the phrase they head. What is crucial in language is that the head of a phrase often dictates the way in which its dependents are pronounced (e.g. adjectives agree with nouns in certain languages). In our model we have not generated the head-parameter as such, since we are only encoding conditional probabilities and not linear order, but we have managed to model dependency of expression. The way a property is communicated is conditional on the object.

# 7 Conclusion

This paper has explored different variations on the Naming Game in a complex world. We have shown that different biases interact with the Naming Game, sometimes generating surprising results. One of our simulations has generated something similar to grammatical dependency from an essentially unrelated bias. The Naming Game continues to serve as an interesting tool for investigating the emergence of human language.

# References


Chomsky, N. and Lasnik, H. (1993). Principles and parameter theory. In von Stechow, A., Sternefeld, W., and Vennemann, T., editors, *Syntax: an International Handbook of Contemporary Research*, pages 506–569. Walter de Gruyter, Berlin.

Gouet, R. (1997). Strong convergence of proportions in a multicolor Pólya urn. *Journal of Applied Probability*, 34(2):426–435.

Kaplan, F., Steels, L., and McIntyre, A. (1998). An architecture for evolving robust shared communication systems in noisy environments. In *Proceedings of Sony Research Forum 1998*.