Lexical Redundancy, Naming Game and Self-constrained Synonymy

Kerem Eryılmaz (kerem@ii.metu.edu.tr)
Cognitive Science, Middle East Technical University (METU), Ankara 06800, Turkey

Cem Bozşahin (bozsahin@metu.edu.tr)
Cognitive Science, METU, Ankara 06800, Turkey

Abstract

Language games are tools to model some aspects of the social aspects of language and communication. Our approach aims to cover the ground between the elementary naming game and the complex models for social use, for the growth of possibly redundant community and personal lexicons. It uses weighted lists of words for the personal lexicon, probabilistic choice as a selection mechanism and lateral inhibition as the weight update scheme. The results demonstrate that the model is a generalization of the elementary naming game, and it provides a good picture of how big a lexicon agents use for the task, how this size can be controlled using the model parameters and a possible way of explaining how synonymy is kept under control.

Keywords: emergence; stochastic naming game; synonymy; language game; semiotic dynamics

Introduction

Language is a social phenomenon. Although there are uses to language that are not social, and there are modes of social interaction other than language use, a very clear and salient function of language is collaboration and communication among individuals in a population. However, computational studies on language until mid-90s had been more focused on modelling the individual capacities in language acquisition and performance, and did not provide much insight into what language might look like from the viewpoint of a population of language users.

A new generation of research addresses this problem by investigating language as it is produced, used and propagated in a community of language users. A fruitful approach has been to see language as a complex adaptive system. What this basically implies is that only at the population level can we adequately characterize language use, and only by taking into account that language is constantly reshaped by its users. How an individual uses language and how the population generally uses language can and do affect one another. In short, the claim is that some features of language do not stem directly from linguistic or cognitive capacity but from a society of interacting agents. One such feature is the lexicon and its emergence as a common vocabulary of a community.

This new branch of research proved to be quite fertile and produced a large body of work. One path of this research is exemplified by Kirby, who approaches language as "the result of an interaction between three complex adaptive systems that operate on different timescales: the timescale of biological evolution (phylogeny), the timescale of individual learning (ontogeny) and the timescale of language change (glossogeny)" (Kirby, 2002).

Another avenue is that of Steels, who also sees language as a complex adaptive system (Steels, 2000). His work started as an investigation of semiotic dynamics in simple language games played among artificial autonomous agents, and eventually led to an elaborate construction grammar formalism called Fluid Construction Grammar (FCG) for simulating a population of language users who bootstrap their own language (Steels & De Beule, 2006).

Steels's original models were kept elementary to make them easier to analyze. FCG, however, is quite comprehensive. We feel that there is much to be discovered at various levels of complexity, from the trivial, original "naming games" to the full-blown grammatical complexity of FCG. For example, one feature, the negotiation for names of objects, remains simple: if two agents agree on a name, they agree to adopt that one and discard the alternatives, therefore their lexicons are nonredundant if and when convergence arises. Since complex adaptive systems are very unpredictable in terms of how things might change if we change the structure of interactions, we think it is important that we explore much of the degrees of freedom as we can, trying to characterize the emerging structures and phenomena at each level of complexity added.

In that spirit this paper presents an investigation of a naming game among artificial agents with a slightly more complex, weight-based lexicon scheme rather than the original, and a consequent probabilistic word selection scheme. Our focus here is on the tension between lexicon growth and synonymy in the population.The paper first describes the original naming game. Our extension is described next. Our experiments and results are then presented.

The Original Naming Game

The naming game focuses on vocabulary formation and agreement in a population. The agents try to bootstrap a vocabulary of (proper) nouns that they associate with the objects they try to name (De Vylder & Tuyls, 2006). It is assumed that the agents already know how to send and receive signals, and possess the motivation to do so. It is further assumed that the objects are uniquely identifiable by all agents, so feature sets as in the discrimination game are not employed.[1] The final assumption is that the agents have an independent channel of communication (such as pointing) through which they can reveal their word-object pairings to each other.

[1] Note that it is perfectly possible to combine the two games, as done by Steels (2003).

Formally, the game involves a set of objects $O = \{o_1, ..., o_m\}$, and a set of agents $A = \{\alpha_1, ..., \alpha_n\}$. Each agent a_j possesses a lexicon $L_{a_j} = \{E_{a_j,1}, ..., E_{a_j,k}\}$ where $k \leq m$, and each entry in the lexicon consists of a list of words associated with the object o_i, $E_{a_j,i} = \{w_{i,1}, ..., w_{i,q}\}$. All agents also possess a function that maps from the global set of objects to the agent's repertoire of words ($\phi_a : O \mapsto E_{a_j}$). [2] Therefore, an agent is characterized only by its lexicon. A game consists of these steps:

1. Two agents are chosen, one as the hearer (α_h) and one as the speaker (α_s).
2. The speaker chooses an object (o_i) to refer to, and points to it (i.e. makes his choice explicit without using the system we try to bootstrap)
3. The speaker chooses a word in the lexicon for the object ($\phi(o_i)$), or creates one if necessary.
4. The hearer tries to decode the word into the object being referred to ($w_{k,j} \in E_{h,k}$), and uses the indepedent channel to signal this to the other (e.g. points to it).
5. The speaker agent assesses if the response complies with its lexicon, and makes the hearer aware of this assessment. Agents update their lexicons accordingly.[3]
6. If all agents have identical lexicons, the game stops.

Each game results in success ($\exists w_{i,n} \mid w_{i,n} = \phi_s(o_i); w_{i,n} \in E_{h,k}$) or failure ($\nexists w_{i,n} \mid w_{i,n} = \phi_s(o_i); w_{i,n} \in E_{h,k}$). Upon success, both agents purge their entries for that object of all but the successful word. Otherwise, the hearer adds the new word to its entry of the object, or creates one if necessary. There is no intermediary between a word exchange being successful and a word dominating an agent's lexical inventory for an object; it operates on an all-or-nothing basis.

The Stochastic Naming Game

The current proposal has four key differences from the original model:

1. **Lexical entries**: The lexical entries in the original model are simple lists of words. The proposed model implements lexical entries as *weighted* lists of words, updated upon interactions. This allows a graded behaviour in which words are preferred, and constitutes a more realistic situation in which convergence should be achieved, compared to plain lists.

   More formally, for each agent α_i, an additional value function θ_{α_i} is added to retrieve the weight:

   $$\theta_{\alpha_i} : w_{k,q} \mapsto \mathbb{R}$$

   Basically, this is just a lookup for the weight associated with the word in an agent's lexicon. By adding this, the characterization of an agent is a tuple of the lexicon and α_i, instead of only the lexicon as in the original game:

   $$\alpha_i :< L_{\alpha_i}, \theta_{\alpha_i} >$$

2. **Word selection**: The word selection scheme in the original model simply picks a word from the set of words present in the lexicon. It does not specify how to pick the word. Although there are some suggestions for schemes that optimize convergence (Baronchelli, DallAsta, Barrat, & Loreto, 2005), there is no set practice. The proposed model has a specific scheme that makes a weighted, probabilistic choice of the word to use at each round. This introduces a number of advantages. First, it introduces some noise by not guaranteeing the leading word to be chosen at every round. Some level of noise is often beneficial to convergence in dynamical systems. Second, it is a more realistic scheme, especially when top words have similar scores. Third, it makes the system more fault tolerant by minimizing the impact of successful rounds caused by words that are ultimately going to fail and of unsuccessful rounds caused by words that are ultimately going to succeed.

   Formally, the function ϕ_{α_i} is changed to return a word by a weighted random choice. To this end, we first define a probability distribution P where:

   $$P(w_{k,q}) = \frac{\theta_{\alpha_i}(w_{k,q})}{\sum_y \theta_{\alpha_i}(w_{k,y})} \quad (1)$$

   Subsequently, the word ϕ_{α_i}'s returns can be characterized as a random variable X with distribution P.

   $$\phi_{\alpha_i}(o_k) = X \quad (2)$$

   for which:

   $$X \sim P; X \in E_{\alpha_i,k} \quad (3)$$

3. **Parameters**: As a consequence of the update scheme, there are more parameters in this version of the game than the original one. In particular, three δ-values (δ_{success}, δ_{failure} and $\delta_{\text{inhibition}}$) are added for use in updating the lexicon, whose precise roles are elaborated in the following paragraphs. Additionally, two θ-values (θ_{max} and θ_{min}) are added as the maximum and minimum values for any score in the lexicon.

4. **Update scheme**: The agents in the proposed model no longer discard the competing synonyms (i.e. the other words in the lexical entry) upon a successful interaction. Instead, the agents update the weights of their lexical entries for the object upon every interaction.

   A function ω is added to each agent which returns a new, updated weight function after an interaction:

   $$\omega : \theta_{\alpha_i} \mapsto \theta'_{\alpha_i}$$

[2] The function ϕ_a returns E_a which is a list of *all* the words used for an object from the perspective on agent a, and not the most successful word.

[3] Note that this assessment does not require global knowledge; it is a local decision determined solely by agent's lexicon and its assessment of the interaction with another agent.

This function ω adds or subtracts from scores some predefined δ_{success}, δ_{failure} and $\delta_{\text{inhibition}}$, based on lateral inhibition (Lenaerts, Jansen, Tuyls, & De Vylder, 2005). Upon a successful interaction with word $w_{k,p}$, this function returns a new function θ'_{α_i} and optionally modifies the lexicon of the agent. The modification is that if the resulting score for a word is less than a predefined value θ_{min}, that word is removed from the lexical item for that object. Also, there is a set limit θ_{max} on how large the weight may grow, at which point no weight is added. More formally, ω returns the following upon success:

$$\theta'_{\alpha_i}(w_{k,q}) = \begin{cases} min(\theta_{\alpha_i}(w_{k,q}) + \delta_{success}, \theta_{max}) & if q = p \\ \theta_{\alpha_i}(w_{k,q}) - \delta_{inhibition} & if q \neq p \end{cases} \quad (4)$$

where

$$\omega(\theta_{\alpha_i})(w_{k,q}) = \theta'_{\alpha_i}(w_{k,q}) \quad (5)$$

and the following upon failure:

$$\theta'_{\alpha_i}(w_{k,q}) = \begin{cases} \theta_{\alpha_i}(w_{k,q}) - \delta_{failure} & if q = p \end{cases} \quad (6)$$

where

$$\omega(\theta_{\alpha_i})(w_{k,q}) = \theta'_{\alpha_i}(w_{k,q}) \quad (7)$$

It then modifies the lexicon as follows:

$$L'_a = (L_a / E_{\alpha_i,k}) \cup E'_{\alpha_i,k} \quad (8)$$

where

$$E'_{\alpha_i,k} = \{w | \theta'_{\alpha_i}(w) \geq \theta_{min}; \forall w \in E_{\alpha_i,k}\} \quad (9)$$

With this scheme, it is possible to mimick the original model of Steels by using δ_{success}, δ_{failure} and $\delta_{\text{inhibition}}$,values of 10.0, 0.0, 10.0 respectively. Informally, this makes sure that success always maximizes the weight of a word and always eliminates other synonyms, and that failure does not have an impact on the weights. This, in effect, is the behaviour of the original model.

Methodology

Each parameter set, that is, a tuple of (δ_{success}, δ_{failure}, $\delta_{\text{inhibition}}$) was considered a unique case, and the simulation was run 50 times for each case, using 50 agents and 2 objects.[4] The model is considered to have reached convergence when there is 100% success over a success window of 100 rounds or when it reaches the limit for number of rounds, which is chosen as 500,000 for this study. This is the product of the number of agents and the number of objects, making it very likely that all agents will have taken part in at least one interaction regarding each object, making the success window more meaningful. Also, note that our concept of "convergence" is different from what is used in the literature. It does not mean that all lexicons are identical, it simply means that all lexicons are "similar enough" for the intended purpose, "similar enough" defined as above. This way, it is possible to see if synonyms can exist at the point where the success of any given communication is almost certain.

The model was run with various δ parameters. The method of choosing them was fixing a set of ratios in the form δ_{failure}:δ_{success} and $\delta_{\text{inhibition}}$:$\delta_{\text{failure}}$, and then producing the actual δ values by choosing a value for δ_{success} and calculating the rest using that chosen value.

There were five values for δ_{success} denoted by the set $\{1.0, 3.0, 5.0, 8.0, 10.0\}$. For the ratio δ_{failure}:δ_{success}, the ratios picked were *0.0:1.0*, *0.5:1.0*, *1.0:1.0*, *1.5:1.0* and *2.0:1.0*. The ratios used for $\delta_{\text{inhibition}}$:$\delta_{\text{failure}}$ were *0.0:1.0*, *0.5:1.0*, *1.0:1.0* and *1.5:1.0*. If both δ_{failure} and $\delta_{\text{inhibition}}$ are 0.0 for a case, it is not possible to calculate $\delta_{\text{inhibition}}$ from the ratio, so for those cases $\delta_{\text{inhibition}}$ was set to δ_{min} to provide some negative feedback to the model so that it can converge.

The cases in which $\delta_{inhibition} > 1.25 \times \delta_{failure}$ are excluded since this corresponds to the vicinity of the original model and our aim is at exploring different areas of the parameter space. Only the case represented by the tuple (10.0, 0.0, 10.0) is included for comparison since it exactly corresponds to the behaviour of the original model.

The values δ_{max} and δ_{min} were fixed at 10.0 and 0.1, respectively.

Results

The results make it clear that choosing the original model parameters is not the only viable option for our model. In fact, of the total of 70 parameter sets, only 16 performed worse than the original model parameters in terms of time of convergence.

In the following, we will present the results on how model parameters interact. We are not going to present all the results because of space considerations. The analysis will be made both in terms of time of convergence, relative convergence rate and lexicon size. Relative convergence rate is defined as the time it takes for the system to converge once the average size of the lexicons are maximized (this time point of maximum lexicon size is represented as t_{max} in the simulation). Time of convergence is the total number of rounds from the start of the simulation to converge. Lexicon size is the total number of words in an agent's lexicon.

The results confirm that $\delta_{\text{inhibition}}$ functions as a way to shrink the lexicon to have a greater relative convergence rate, that is, to reduce the time it takes for the convergence to be reached once t_{max} is reached. The functions of δ_{success} and δ_{failure} are straightforward as positive and negative feedback,

[4] Originally, up to 5 objects were going to be tested but as far as we could tell from the pilots, this did not provide any interesting insights that 2 objects did not provide for our intentions. Since more objects dramatically increased the simulation time and analysis, the simulations were run using just 2 objects.

respectively.

Interaction of $\delta_{success}$ and $\delta_{failure}$

The interaction of $\delta_{success}$ and $\delta_{failure}$ is easier to explain. These are directly counteracting forces, and therefore if $\delta_{failure}$ is greater than $\delta_{success}$, the system fails to converge save a few exceptional cases. This is not surprising since the system, especially before t_{max}, mostly learns by failing the exchanges therefore learning new words. If the impact of failure is higher than that of success, then it becomes really difficult to disseminate some words to all agents so that later they can converge to that word. Basically, they all get discarded before their commonality causes them to become successful across many interactions.

This effect is further compounded by non-zero $\delta_{inhibition}$ values, which, upon occasional initial success, acts effectively as a failure penalty for all non-successful words, further decreasing the number of common words. However, there are a few cases where models with $\delta_{success} \leq \delta_{failure}$ actually converge to a common vocabulary. In a closer look, there are two conditions under which convergence occurs. One is when all δ values are quite small, so that success can accumulate to save at least a couple of alternatives for a word from being discarded. In contrast, larger values mean that one or two failures guarantee discarding of a word, and the number of successes have little impact on this since the scores stop increasing once they hit the upper limit of θ_{max}. In other words, small δ values give the agents some room to keep a greater amount of interaction history, and therefore they become better at evolving their lexicon in tandem with the population trends.

The other case is where $\delta_{inhibition} = 0$. This leaves only $\delta_{success}$ and $\delta_{failure}$ to battle each other, and occasionally leaves room for convergence unless $\delta_{failure} < 1.25 \times \delta_{success}$ or $\delta_{failure} > 0.75 \times \theta_{max}$. This is equivalent to saying that $\delta_{failure}$ should leave room for at least two failures until a previously successful word is discarded, i.e. the word is not discarded right away upon failure. Since there is no other mechanism to decrease the weights, this allows some dissemination with a bit of luck.

Interplay of $\delta_{failure}$ and $\delta_{inhibition}$

The key to the relationship between $\delta_{failure}$ and $\delta_{inhibition}$ is that they can replace one another with slightly different effects. $\delta_{inhibition}$ is a stronger form of $\delta_{failure}$ which affects not only one but almost all (save the successful one for the round) words associated with an object each round. In fact, this is why the original model, with the parameter set (10.0,0.0,10.0), can function without $\delta_{failure}$.

This great impact of $\delta_{inhibition}$ effectively shrinks the lexicon, and this makes the t_{max} smaller but also reduces the relative convergence rate. The reason is that at t_{max} there are less alternatives that the system may converge to for an object. This means that any perturbation in the system, such as an interaction where the speaker agent prefers a not-to-be-successful word, needs to be counteracted so that the system returns to moving towards the word it originally had been converging to.

In contrast, a big lexicon at t_{max} means there are many alternatives, and a disturbance need not be fully counteracted; the system might just converge to another word-object pairing that is salient in the population. Accordingly, our results show that a relatively large $\delta_{failure}$ combined with a small $\delta_{inhibition}$ produces the fastest convergences, with a large lexicon at t_{max} since $\delta_{inhibition}$ does not get a chance to shrink the lexicons as much. After that, applications of $\delta_{inhibition}$ mostly help convergence to the pairings that will ultimately dominate.

Lexicon Size

Lexicon size can be used as an indicator of game dynamics. Since it is not a parameter but a quantity that manifests itself during the game, it is difficult to test. Nonetheless, there are some clear tendencies that are important.

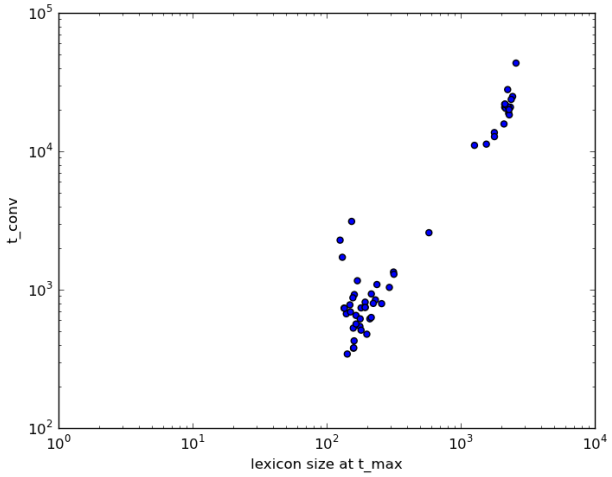

Figure 1: A log-log plot of time of convergence versus average lexicon size for all convergent parameter sets.

The first such tendency is between the time of convergence and average lexicon size at t_{max}. It turns out that there is a direct proportion between the logarithm of average lexicon size and the logarithm of the time of convergence. Roughly, this means the bigger the lexicon size at t_{max}, the longer the convergence takes. This indicates that the time of convergence and relative convergence rate are not necessarily correlated. Although parameter sets that produce bigger lexicons converge fast after t_{max}, they also take longer to put together (i.e. to reach t_{max}) and therefore do not necessarily represent the sets that allow convergence in the minimum number of rounds.

The second tendency is best represented by a plot of rounds (i.e. the time series) versus average lexicon size (see Figure 2). The plots fork into two fairly distinct groups, and all of the plots with bigger lexicon size belong to parameter sets in

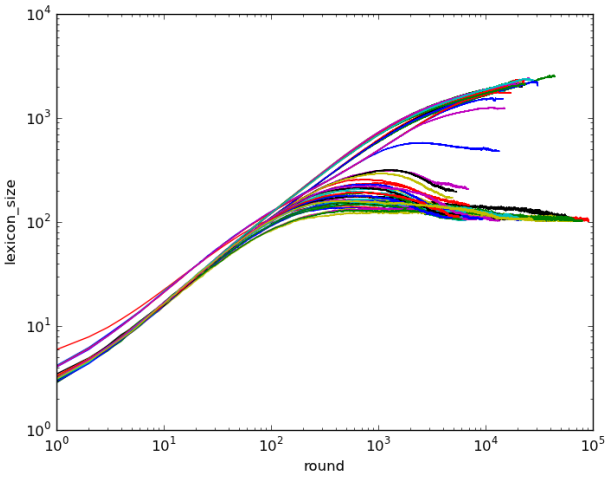

Figure 2: A log-log plot of time of convergence versus average lexicon size for all convergent parameter sets.

which $\delta_{\text{inhibition}} \leq 0.1$. This demonstrates that $\delta_{\text{inhibition}}$ has the function of shrinking the lexicon. In the figure, it is apparent that for all cases, there is a peak (whose position in time we call t_{max}) after which convergence is achieved (which is also a finding encountered in the original naming game literature). For very small values of $\delta_{\text{inhibition}}$, this peak is more or less identical to the time point at which convergence is achieved. In other words, $\delta_{\text{inhibition}}$ shrunk the lexicon so little that the system converged with a larger average lexicon size. In other words, on average, at least some of the objects have more than one alternative word associated with them although the success rate climbed to 100% over a window of 100 rounds. This may hold an insight into how synonymy may be controlled in human languages.[5] In our model, synonymy does not necessarily hinder mutual communication. In more complex models, if they have analogous tendencies at all, it might be beneficial for large populations to be able to communicate very successfully without a full agreement on which object is referred to by which word.

This also suggests that memory requirement for this task is not necessarily determined by the amount of memory available to the agents but by the type of learning a population adopts (or is born with), and the fact that the task is undertaken by a population. Even if they had very big memory capacities (i.e. more than they need to use), part of this capacity would be redundant because of the decay in weights of the unsuccessful or non-successful words would get them discarded from the lexicon.[6] However, parameter sets with small $\delta_{\text{inhibition}}$ (similar to those in the top partition of Figure 2) can conceivably make use of more memory than available here. This indicates that $\delta_{\text{inhibition}}$ is a way of controlling the memory use of the population for the task.

[5]Of course, these are results from a single study with very strict and narrow boundaries in terms of what it deems possible "language"; hence the verb *inspiration* instead of *insight*.

[6]Non-successful words are those whose weights are decremented not because they failed, as unsuccessful words' are, but because some other word succeeded and $\delta_{\text{inhibition}}$ was nonzero.

Conclusion

We believe that this line of research is quite relevant to cognitive science and to study of linguistics in general. Although the communicative capacities of our agents are very limited in that they do not contain any capabilities for syntax, discourse etc., our results are somewhat reminiscent of Elman's work on importance of starting with a small working memory in language acquisition (1993). Our model shows that using a reasonable $\delta_{\text{inhibition}}$ actually facilitates learning although it does reduce the amount of memory used for the task. In our model, it is not even about the availability of more memory; the memory simply does not need to be larger.

This would not make sense for learning static knowledge where having as many samples as possible at any point in time is the goal. But such a finding is arguably not as surprising for learning population-generated knowledge which may change from one point in time to the other. The memory constraint comes not from the learners but from the nature of what is learned, how it changes and the relationship between the learners and what is learned. The constraint is not on any given specific lexicon but on the "lexicon of the population". In other words, this effect does not really stem from the agents themselves but from the fact that what they are trying to agree on is malleable by this very process of negotiation.

It also confirms the previous findings in the semiotic dynamics literature that the sudden popularity of a word upon coinage is an illusion. The conclusion from this literature, including this study, is that there are often many competing synonyms for the same concept, and, after a word hits a popularity threshold, the system falls into a spiral of making the most popular word even more popular. This is even true in a model such as ours where there is a net negative feedback, δ_{failure}, and varying ratios of the strength of other mechanisms of pressure, unlike other models in the literature. There are grey areas where words are neither in disuse nor popular, and they need not become either unused or popular for convergence to occur.

Finally, these simulations can give us an idea about controlling synonymy. Our model does not necessarily have much pressure towards eliminating synonymy; it just works by looking at how successful communications are. The only pressure on synonymy is controlled by $\delta_{\text{inhibition}}$. Such inhibition effectively means that success of one word for an object is the failure of others for that same object. However, unlike simpler models in the literature, synonymy can be preserved while still achieving very high communicative success, though only when using low $\delta_{\text{inhibition}}$ values and with the side effect of having a larger vocabulary to hold the synonyms. The fact that the model also converges with multiple synonyms for an object both when $\delta_{\text{inhibition}} \neq 0$ and $\delta_{\text{inhibition}} = 0$ demonstrates that this ability to maintain syn-

onymy is not about the existence but the magnitude of the pressure towards no synonymy, as exerted by $\delta_{\text{inhibition}}$.

The main lesson to learn from this is that synonyms are eliminated only if they seriously hinder communicative success, and they are eliminated more or less globally when they do. It would be more feasible to use synonyms if the agents had contextual cues as humans do to constrain the search space, but synonymy can be preserved even without this additional information. If the ambiguity they bring is manageable within the task we are trying to achieve with our language (and these models demonstrate that there really exist such tasks), they need not be eliminated. This is analogous to the difference between colloquial text and legislation in terms of ambiguity, which arguably has something to do with what the population of language users intend to use the language for. What we are trying to achieve in such tasks is not necessarily identical lexicons but communicative success, and what this means has to be defined on a per-case basis.

Figure 2 shows that we pay the price for synonymy if we let it loose: it causes either late convergence or no convergence. The natural equivalent of these results are open to discussion. Our suggestion is that they seem to indicate keeping it low among a community while not completely eliminating it seems to be a way of balancing expressivity and communicative success. If either aspect begins to dominate, it will be equivalent to widespread synonymy and no synonymy, respectively. The conclusion that these may be counteracting forces are shown by synonym's relation to number of iterations, rather than assumed cognitively.

References

Baronchelli, A., DallAsta, L., Barrat, A., & Loreto, V. (2005). Strategies for fast convergence in semiotic dynamics. *Word Journal Of The International Linguistic Association*, 6.

De Vylder, B., & Tuyls, K. (2006). How to reach linguistic consensus: A proof of convergence for the naming game. *Journal of theoretical biology*, *242*(4), 818–831.

Elman, J. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*.

Kirby, S. (2002, April). Natural Language From Artificial Life. *Artificial Life*, *8*(2), 185–215.

Lenaerts, T., Jansen, B., Tuyls, K., & De Vylder, B. (2005). The evolutionary language game: An orthogonal approach. *Journal of Theoretical Biology*, *235*(4), 566–582.

Steels, L. (2000, December). language as a complex adaptive system. In M. Schoenauer et al. (Eds.), *Parallel problem solving from nature-ppsn vi* (Vol. 6).

Steels, L. (2003, July). Evolving grounded communication for robots. *Trends in Cognitive Sciences*, *7*(7), 308–312.

Steels, L., & De Beule, J. (2006). A (very) brief introduction to fluid construction grammar. In *Proceedings of the third workshop on scalable natural language understanding scanalu 06* (p. 73). Association for Computational Linguistics.