Meaning Space Structure Determines the Stability of Culturally Evolved Compositional Language

Henry Brighton (henryb@ling.ed.ac.uk)
Language Evolution and Computation Research Unit,
Department of Theoretical and Applied Linguistics,
The University of Edinburgh, Edinburgh, UK

Simon Kirby (simon@ling.ed.ac.uk)
Language Evolution and Computation Research Unit,
Department of Theoretical and Applied Linguistics,
The University of Edinburgh, Edinburgh, UK

Abstract

Explanations for the evolution of compositional and recursive syntax have previously attributed these phenomena to the genetic evolution of the language acquisition device. Recent work in the field of computational evolutionary linguistics suggests that syntactic structure can instead be explained in terms of the dynamics arising from the cultural evolution of language. We build on this previous work by presenting a model of language acquisition based on the Minimum Description Length principle. Our Monte Carlo simulations show that the relative cultural stability of compositional language versus non-compositional language is greatest under conditions specific to hominids: a complex meaning space structure.

Introduction and Related Work

Human language differs greatly from other natural communication systems. Our use of compositional and recursive syntax places us in a unique position: we can comprehend and produce an ostensibly infinite number of utterances. Why are we alone in this position? Human language is a result of three adaptive systems: learning, genetic evolution, and cultural evolution. Over the past half century cognitive scientists has addressed the problem of learning. The past ten years has seen a resurgence of interest in the evolutionary basis of language (Pinker & Bloom, 1990). Only recently has the cultural evolution of language been seriously analysed. Hare & Elman (1994) outlined perhaps the first *iterated learning model*. The iterated learning model seeks to model the evolution of language through generations of language users, solely on the basis of each agent observing the behaviour of the previous generation (Kirby, in press b). Recent demonstrations of the cultural evolution of compositionality and recursive syntax (Kirby, in press a; Batali in press) suggest that these properties of human language, traditionally attributed to genetic evolution, can in fact be explained as emergent properties arising from the dynamics of iterated learning. One criticism levelled at these models is that the learning bias of the individual agents is typically too strong – the observed behaviour is striking yet inevitable (Tonkes & Wiles, in press).

Here, we consider compositional syntax – the property of human language whereby the meaning of a signal is some function of the meaning of its parts. We address the criticisms of bias strength by employing the Minimum Description Length (MDL) principle, which rests on a solid mathematical justification for induction. We demonstrate that the relative stability of compositional language, with respect to non-compositional language, is at a maximum under two conditions specific to hominids: (a) a complex meaning space, and (b), limited language exposure, a situation commonly referred to as the *poverty of the stimulus*. Gell-Mann (1992) was perhaps the first to suggest the relevance of Kolmogorov Complexity, which is closely related to MDL, to the study of language evolution. Our use of the MDL principle is similar to that of Teal et al (1999), who model change in signal structure using the iterated learning model. Our model extends this work by considering the role of meanings, as well as allowing signals of arbitrary length. The structure of this article as follows. First, we outline the MDL principle and introduce a novel hypothesis space. We then discuss issues of stability and learnability in the context of cultural evolution. Finally we illustrate the impact of meaning space complexity on the stability of compositional language. Our main goal is to establish properties of compositional language relative to a more sound model of linguistic generalisation.

Hypothesis Selection by MDL

Ranking potential hypotheses by minimum description length is a highly principled and very elegant approach to hypothesis selection (Li & Vitányi, 1997). The MDL principle can be derived from Bayes's Rule, and in short states that the best hypothesis for some observed data is the one that minimises the sum of (a) the encoding length of the hypothesis, and (b) the encoding length of the data, when represented in terms of the hypothesis. A tradeoff then exists between small hypotheses with a large data encoding length and large hypotheses with a small data encoding length. When the observed data contains no regularity, the best hypothesis is one that represents the data verbatim, as this minimises the data encoding length. However, when regularity does exist in the data, a smaller hypothesis is possible which describes the regularity, making it explicit, and as result the hypothesis describes more than just the observed data. For this reason, the cost of encoding the data increases. MDL tells us the ideal tradeoff between the length of the hypothesis encoding and the length of the data encoding described relative to the hypothesis. More formally, given some observed data D and a hypothesis space H the best hy-

pothesis h_{MDL} is defined as:

$$h_{MDL} = \min_{h \in H} \{L_{C_1}(h) + L_{C_2}(D|h)\} \quad (1)$$

where $L_{C_1}(h)$ is the length in bits of the hypothesis h when using an optimal coding scheme over hypotheses. Similarly, $L_{C_2}(D|h)$ is the length, in bits, of the encoding of the observed data *using* the hypothesis h. We use the MDL principle to find the most likely hypothesis for an observed set of meaning/signal pairs passed to an agent. When regularity exists in the observed language, the hypothesis will capture this regularity, when justified, and allow for generalisation beyond what was observed. By employing MDL we have a more theoretically solid justification for generalisation. The next section will clarify the MDL principle – we introduce the hypothesis space and coding schemes.

The Hypothesis Space

We introduce a novel model for mapping strings of symbols to meanings, which we term a Finite State Unification Transducer (FSUT). This model extends the scheme used by Teal et al (1999) to include meanings and variable length strings. Given some observed data, the hypothesis space consists of all FSUTs which are consistent with the observed data. Both compositional and non-compositional languages can be represented using the FSUT model.

Throughout this paper, a meaning is defined as a set of features represented by a vector, with each feature taking a value. A *meaning space profile* describes the structure of a meaning space. For example, the meaning space profile $(3,3)$ defines a meaning space with two dimensions, each dimension having three possible values. Signals are just strings of symbols (of arbitrary length) drawn from some alphabet Σ. A Finite State Unification Transducer is specified by a 6-tuple $(Q, \Sigma, \Omega, \delta, q_0, q_F)$ where Q is the set of states used by the transducer, Σ is the alphabet from which symbols are drawn, and Ω is the meaning space profile which defines the structure of the meaning space. The transition function δ maps state/symbol pairs to a new state, along with the (possibly under specified) meaning corresponding to that part of the transducer. Two states, q_0 and q_F need to be specified, they are the initial and final state, respectively. Consider an agent A, which receives a set of meaning/signal pairs during acquisition. For example, a simple observed language might be the set:

$$L_1 = \{(\{2,1\}, \text{cdef}), (\{2,2\}, \text{cdgh}), (\{1,2\}, \text{abgh})\}$$

Figure 1(a) depicts an FSUT which models L_1. We term this transducer the *prefix tree transducer* – the observed language and only the observed language is represented by the prefix tree transducer. The power of the FSUT model only becomes apparent when we consider possible generalisations made by merging states and edges in the transducer. Figure 1(c) shows a compressed transducer. Here, the some of the states and meaning labels attached to the edges in the prefix tree transducer have been merged. There are two merge operations:

1. *State Merge*. Two states q_1 and q_2 can be merged if the transducer remains consistent. All edges that mention q_1 or q_2 now mention the new state.
2. *Edge merge*. Two edges e_1 and e_2 can be merged if they share the same source and target states and accept the same symbol. The result of merging the two edges is a new edge with a new meaning label. Meanings are merged by finding the intersection of the two component meanings. Those features which do not have values in common take the value ? – a wildcard which matches all values. As fragments of the meanings may be lost, a check for transducer consistency is also required.

Figure 1(b) illustrates which state merge operations are applied to the prefix tree transducer in order to compress it. Figure 1 is simple example, as the resulting transducer does not generalize: only the observed meaning/signal pairs can be accepted or produced.

Encoding Lengths

In order to apply the MDL principle we need an appropriate coding scheme for: (a) the hypotheses, and (b) the data using the given hypothesis. These schemes correspond to $L_{C_1}(h)$ and $L_{C_2}(D|h)$ introduced in Equation 1. The requirement for the coding scheme L_{C_1} is that some machine can take the encoding of the hypothesis and decode it in such a way that a unique transducer results. Similarly, the coding of the data with respect to the transducer must describe the data uniquly. To encode a transducer $T = (Q, \Sigma, \Omega, \delta, q_0, q_F)$ containing n states and m edges we must calculate the space required, in bits, of encoding a state ($S_{state} = log_2(n)$), a symbol ($S_{symbol} = log_2(|\Sigma|)$), and a meaning ($S_{meaning} = \sum_{i=1}^{|\Omega|} log_2(\Omega_i + 1)$). The encoding length of the transducer is then:

$$S_T = m\{2S_{state} + S_{symbol} + S_{meaning}\} + S_{state}$$

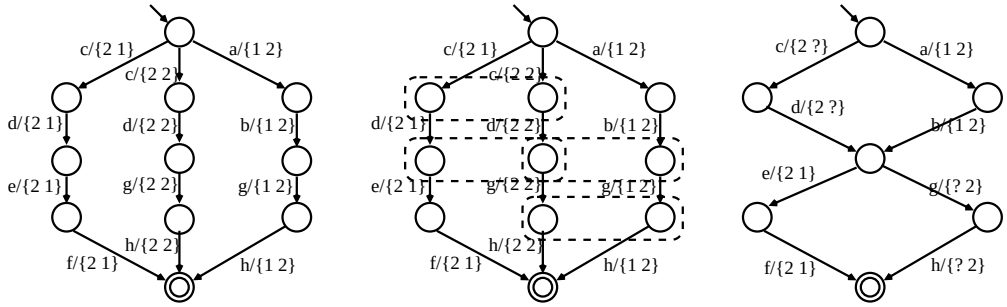
which corresponds to encoding the transition function δ along with the identity of the accepting state. To enable the machine M to uniquely decode this tranducer we must also specify the lengths of constituent parts. We term this part of the encoding the *prefix block*:

$$S_{prefix} = S_{state} + 1 + S_{symbol} + 1 + S_{meaning} + 1$$

To calculate $L_{C_1}(h)$ we then use the expression:

$$L_{C_1}(h) = S_{prefix} + S_T \quad (2)$$

The data encoding length is far simpler to calculate than the grammar encoding length. For some string s composed of symbols $w_1 w_2 \ldots w_{|s|}$ we need to detail the transition we choose after after accepting each symbol with respect to the given transducer. The list of choices made describes a unique path through the transducer. Additional information is required when the transducer enters



(a) Prefix Tree Transducer (b) Merge Operations (c) Compressed Transducer

Figure 1: (a) The prefix tree transducer. (b) The state merge operations required to induce the compressed transducer shown in (c).

an accepting state as the transducer could either accept the string or continue parsing characters, as the accepting state might contain a loop transition. Given some data D composed of p meaning/signal pairs, $L_{C_2}(D|h)$ is calculated by:

$$L_{C_2}(D|h) = \sum_{i=1}^{p} \sum_{j=1}^{|s_i|} \{\log_2 z_{ij} + F(s_{ij})\} \qquad (3)$$

where z_{ij} is the number of outward transitions from the state reached after parsing j symbols of the ith string. The function F handles the extra information for accepting states:

$$F(s_{ij}) = \begin{cases} 1 & : \text{ when the transducer is in } q_F \\ 0 & : \text{ otherwise} \end{cases}$$

Prefix tree transducers are compressed by applying the merge operators described above. We use a hill climbing search. All the merge operators are applied in turn and the one which leads to the greatest reduction in $L_{C_1}(h) + L_{C_2}(D|h)$ is chosen. The process is repeated until this expression cannot be minimised further.

Iterated Learning

Cultural evolution transmits information down generations by non-genetic means. The cultural evolution of language results from language users inheriting the linguistic behaviour of previous generations. We model this process using the Iterated Learning Model (Kirby, in press b). Each generation consists of a single agent which observes the linguistic performance of the agent in the previous generation. This process is repeated over (usually) thousands of generations. Under conditions of perfect transmission, the language of each generation would be identical. This is not how human language works, as real language users suffer from the *poverty of the stimulus*: language users only ever see a fraction of possible utterances, yet are capable of producing and comprehending an ostensibly infinite number of utterances. Obviously, language users make generalisations from language they have observed. The sparsity of language exposure is modelled here using a *communication bottleneck*. The bottleneck is imposed on the agents in our simulations by restricting the number of utterances each generation observes. Initially this restriction results in each generation having a different language, the language changes down the generations: we have a dynamic system.

More precisely, the iterated learning model consists of a series of learners which are called on in turn to express a random subset of the meaning space to the next learner in the series. If the meaning space consists of n different meanings, some number m ($m \leq n$) of distinct random meanings are observed by each agent, although the total number of meanings observed may be larger than n as some are repeated. We are interested in language designs that result in stability. For stability to occur the whole mapping from meanings to signals must be recoverable from limited exposure: the language must be *learnable*.

In the experiments that follow we analyse compositionality, the property of language in which the meaning of a sentence is a function of the meanings of its parts. Are compositional language designs stable? We contrast compositionality with non-compositionality, i.e., signals whose meaning is not a function of the meaning of its parts. In our model non-compositional languages are those where the mapping from meanings to signals is random.

Compression and Learnability

The evolving language, as it passes through generations of language users, can be seen as a complex system. We are interested in the nature of steady states – attractors – those languages which are stable and persist. One way of characterising stable languages is in terms of expressivity. If a language can express all possible meanings and is learnable then it will persist. Instead of modelling the full iterated learning model we try and establish the conditions for stability. To do this we construct two languages: some compositional language L_{comp} and some random language $L_{noncomp}$. Through experimentation we identify the learnable transducers which possess the min-

imum description length, described above, for both types of language. We conduct Monte Carlo simulations to establish how expressiveness depends on the structure of the meaning space.

Compositional Languages

To construct a compositional language each feature value is assigned a unique word which is used in forming the signal for meanings containing that feature value. Uniqueness is not a necessity for feature values – values occurring in other features can share the same word. For example, given a meaning space profile $(3,3)$ we could construct the compositional language L_2:

$$\begin{aligned} L_2 \quad = \quad \{ & (\{1,1\},\text{aa}),(\{1,2\},\text{ab}),(\{1,3\},\text{ac}), \\ & (\{2,1\},\text{ba}),(\{2,2\},\text{bb}),(\{2,3\},\text{bc}), \\ & (\{3,1\},\text{ca}),(\{3,2\},\text{cb}),(\{3,3\},\text{cc}) \ \} \end{aligned}$$

For purposes of clarity single symbol words are used in the signal to denote feature values. Variable length words could have been used.

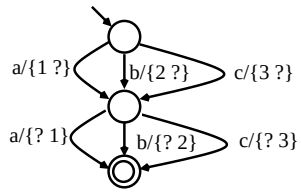


Figure 2: The MDL transducer for languages L_2 and L_3.

Which is the best hypothesis, according to MDL, for this data? Figure 2 depicts the MDL transducer which accepts this language. Consider the language $L_3 = L_2 - \{(\{3,3\},\text{cc})\}$. Exactly the same transducer is induced by MDL when L_3 is observed. The transducer has generalised from the data to account for the missing sentence. L_2 is learnable, even when the learner is not exposed to all the sentences in L_2. The structure of the transducer depicted in Figure 2 is typical for a compositional language. These transducers are learnable from compositional input. From now on we term these transducers *compressed transducers*. Equivalent random languages are constructed by assigning random signals to each meaning. Occasionally the MDL transducer for random languages is smaller than the prefix tree transducer, but correct generalisation cannot occur. Below, we analyse the properties of these two families of transducer.

Meaning Space Structure Affects Learnability and Stability

In this section we investigate how meaning space structure impacts on the learnability and stability of compositional and non-compositional languages. We consider two types of transducer: compressed transducers and prefix tree transducers. By carrying out Monte Carlo simulations, we show how, (a) the size of the communication bottleneck, and (b), the meaning space structure, affects the degree of language stability with respect to the iterated learning model.

Preliminaries

Usually the proportion of the meaning space expressed by an agent at each generation is given by the number of random meanings the agent must express. The following analysis requires a more concrete measure of the degree of exposure to the meaning space. For example, given a meaning space composed of n meanings and a bottleneck size of m, significantly fewer than n distinct meanings will be observed. Below, we measure the bottleneck size in terms of *meaning space coverage* which is just the expected proportion of the meaning space sampled when picking at random. The expected coverage, c, when picking r elements at random (with replacement) from n is $c = 1 - (1 - \frac{1}{n})^r$

Stability

A stable language is one which survives the communication bottleneck – it occurs when a transducer is induced with maximum expressivity. For example, Figure 3(a) shows that, given a compositional language, the compressed transducer reaches maximum expressivity after seeing only 20% of the meaning space. This is because exposure to feature values is required rather than exposure to whole meanings (recall the structure of compressed transducers shown in Figure 2). Maximum expressivity results when all the feature values have been observed, and as a result, induction can account for novel meanings whose individual feature values have already been seen. As one would expect, the expressivity of a prefix tree transducer, the best hypothesis for a non-compositional language, increases linearly with coverage. Here, expressivity depends on the exposure to whole meanings. In order for the entire meaning space to be communicated, an infinitely large bottleneck is required: prefix tree transducers will rarely result in maximum expressivity.

Learnability

Given a compositional language as input which transducer results in the smallest description length? In terms of transducer size, compressed machines will always be smaller, but what influence does the data encoding length have? Figure 3(b) shows the relative size of encoding lengths for the compressed transducer versus the prefix transducer. When the size difference lies above the baseline compressed transducers are chosen, and below the baseline, prefix tree transducers are chosen. Figure 3(b) illustrates that compressed machines are always preferable. This is not the case when we consider *amplified* bottlenecks – where we multiply the frequency of meanings passed through the bottleneck. For future work this observation is relevant, as we intend to investigate different probability distributions over the meaning space. Figure 3(c) shows the results for a 100-fold amplified

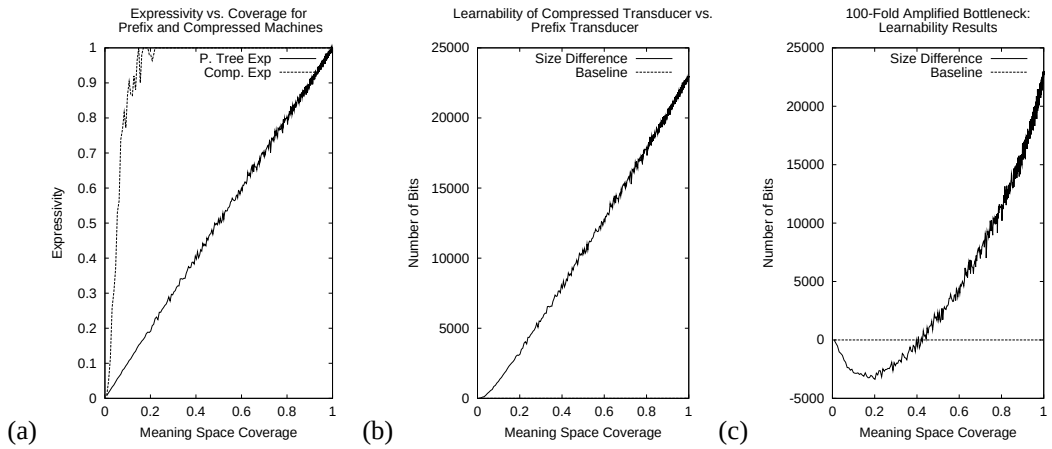


Figure 3: (a) Expressivity as a function of coverage for prefix tree transducers and compressed transducers, (b) depicts the size advantage of compressed machines over prefix tree machines: the MDL search always chooses compressed machines, (c) shows how an amplified bottleneck affects learnability. A meaning space structure of (2,2) was used.

bottleneck. For small coverage values the prefix tree transducer is preferable. The MDL measure prefers the transducer which does not generalise: the less evidence we have of the sample space, the less we are justified in accurately postulating the existence of unseen members of the sample space. This justification becomes weaker the more we see of the sample space. MDL reflects this intuition by preferring compressed transducers at higher coverage values.

Competing Languages

Above, we argued that compositional languages are more stable than non-compositional languages. In short, compression, and as a result generalisation and high expressivity, is only possible with compositional languages. Non-compositional languages, by definition, lack any form of regularity in the mapping between meanings and signals and are therefore far less compressible. We also illustrated that for compositional languages, high expressivity through compression is achievable for low meaning space coverage, as induction via compression is a function of degree of exposure to feature values, rather than whole meanings. This tells us that the size of the bottleneck which maximises the relative stability of compositional language versus non-compositional language is a function of meaning space structure.

Ultimately, we are interested in the question: Under what circumstances is compositional language most likely to occur? But now we can reformulate the question: When is the relative stability of compositional languages versus non-compositional languages at a maximum? The relative stability of a compositional language over a non-compositional language can measured by comparing the expressivity of the transducers chosen by MDL for each language type. For example, given some compositional language L_{comp} we identify the most likely hypothesis on the basis of MDL. This gives us a transducer T_{comp} with expressivity E_{comp}. Similarly, for a non-compositional language $L_{noncomp}$ we identify $T_{noncomp}$ with expressivity $E_{noncomp}$. The relative stability measure tells how much of a stability advantage compositional language provides. We denote this quantity as R and define is as:

$$R = \frac{E_{comp}}{E_{comp} + E_{noncomp}}$$

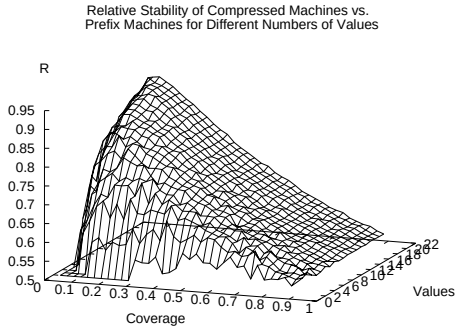


Figure 4: The relative expressivity, R, for a two-feature meaning space for different numbers of values.

Now, constructing compositional languages by fixing the number of features but varying the number of feature values, for different meaning space coverage values, and then measuring R, will provide an insight into how R depends on the meaning space structure. Figure 4 illustrates this dependency. The striking feature of these results is that compositionality is most likely, or more preferable, when the number of values per feature is large and the meaning space coverage is small. The payoff, R,

in the number of values per feature decreases as the number of values increases. Similar results occur when we fix the number of values per feature but increase the number of features. Again, small bottlenecks and many features lead to a large payoff when considering compositional languages. It appears that the more complex the meaning space, the higher the R value, especially for small bottlenecks. However, R does not increase linearly with meaning space complexity, the payoff achieved through increased meaning space complexity deteriorates.

Perhaps a more informative analysis results when we consider the problem of communicating about some fixed number of objects. For example, given the problem of describing 1000 objects, which meaning space structure leads to the occurrence of compositional language being most likely? Figure 5 shows that compositionality is most likely when the 1000 objects are discriminated more by features than by feature values. The greater the number of features, the smaller the number of observations required before all feature values are seen. Only when many feature values have been observed can induction justifiably be applied. However, as before, this payoff does not increase linearly with the number of features. After a point, more features offer little advantage.

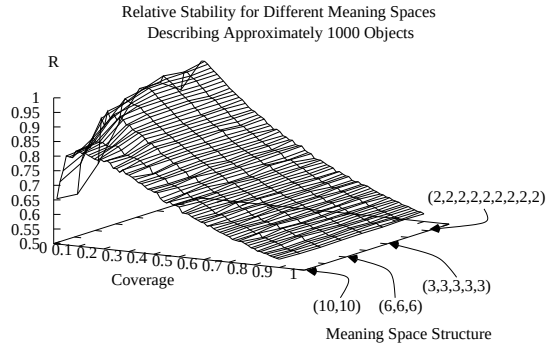


Figure 5: The relative expressivity, R, for different ways of describing approximately 1000 objects.

These results tell us when the stability of compositional language is at a maximum, in comparison to non-compositional language. These conditions provide the first steps of an explanation for the emergence of compositionality in human language. The poverty of stimulus coupled with the supposed complexity of the hominid mind are exactly the conditions under which these experiments predict compositional language is most likely to emerge.

Conclusions

By providing a sound basis for induction we have addressed criticisms of the poorly justified, and arguably overly strong, inductive bias typical of earlier work on the cultural evolution of syntactic language. However, the chief point we aim to make is that compositionality is advantageous under conditions specific to hominids:

1. *Complex meaning space structure:* Hominids carve up their perceived environment into many features, at least, their perception is unlikely to be restricted to holistic experiences.

2. *The poverty of the stimulus:* The need for a communication system with high expressivity is required if meanings drawn from a complex meaning space are to be communicated. However, limited exposure to this mass of possible utterances is all that is required for unlimited comprehension and production.

Using a mathematical model, Nowak, Plotkin, & Jansen (2000) present a similar argument with respect to the genetic evolution of syntactic structure: the complexity of the perceived environment leads to a pressure for syntax. Our argument is similar in spirit, but demonstrates that natural selection is not the only mechanism which can explain the emergence of syntax.

References

Batali, J. (in press). The Negotiation and Acquisition of Recursive Communication Systems as a Result of Competition among Exemplars. In Briscoe, E. (Ed.), *Linguistic Evolution through Language Acquisition: Formal and Computational Models*, Cambridge University Press.

Gell-Mann, M. (1992). Complexity and Complex Adaptive Systems. In Hawkins, J. A. and Gell-Mann, M. (Eds.) *The Evolution of Human Languages*. Addison-Wesley.

Hare, M., & Elman, J. L. (1995). Learning and Morphological Change, *Cognition*, *56(1)*.

Kirby, S. (in press a). Learning, Bottlenecks and the Evolution of Recursive Syntax. In Briscoe, E. (Ed.), *Linguistic Evolution through Language Acquisition: Formal and Computational Models*, Cambridge University Press.

Kirby, S. (in press b). Spontaneous Evolution of Linguistic Structure: An Iterated Learning Model of the Emergence of Regularity and Irregularity, to appear in: *IEEE Transactions on Evolutionary Computation*.

Li, M., & Vitányi, P. (1997). *A Introduction to Kolmogorov Complexity and Its Applications*. New York: Springer-Verlag.

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

Pinker, S., & Bloom, P. (1990). Natural Language and Natural Selection. *Behavioral and Brain Sciences*, *13*.

Teal, T., Albro, D., Stabler, E, & Taylor, C.E. (1999). Compression and Adaptation. *Fifth European Conference on Artificial Life* (pp. 709–719). Springer-Verlag.

Tonkes, B., & Wiles, J. (in press). Methodological Issues in Simulating the Emergence of Language. To appear in a volume arising from: *Third Conference on the Evolution of Language*, Paris, 2000.