Syntax out of Learning: the cultural evolution of structured communication in a population of induction algorithms

Simon Kirby

Language Evolution and Computation Research Unit, Department of Linguistics, University of Edinburgh, Scotland
simon@ling.ed.ac.uk
http://www.ling.ed.ac.uk/~simon

Abstract. A new approach to the origins of syntax in human language is presented. Using computational models of populations of learners, it is shown that compositional, recursive mappings are inevitable end-states of a cultural process of linguistic transmission. This is true even if the starting state is no language at all. It is argued that the way that knowledge of language is transmitted through a learning bottleneck profoundly influences its emergent structure. This approach provides a radical alternative to one in which the structure of language is viewed as an innate, biological adaptation to communicative pressures.[1]

1 The origins of syntax

Human language is unique among natural communication systems in having a compositional and recursive mapping between meanings (mental representations to be communicated) and forms (linear strings of, typically phonetic, gestures). It is also extremely unusual in the way it is learnt. Each generation acquires at least some of the meaning-form mapping by observing the use of the previous generation's mapping. It has been argued that this type of learning of mappings is also unique to humans (Oliphant, 1998). In this paper I will explore, using a working model of linguistic transmission, the links between these two features of language. In particular, I aim to explain the origins of syntax in language by looking at general properties of the transmission of learned behaviour.

The explanation put forward here contrasts radically with what is perhaps the dominant approach to the origins of syntax. In many linguists' view, syntax is more or less fully specified by a learning mechanism (the Chomskyan Language Acquisition Device, or LAD) which significantly constrains the language learner with prior knowledge about the nature of language (Chomsky, 1986). Prior biases or constraints on a learner are assumed to be innate, and therefore genetically determined. This leads naturally for many researchers (see Pinker and Bloom,

[1] This work benefited greatly from conversation with and comments from Jim Hurford, Mike Oliphant, and Ted Briscoe and was supported by ESRC grant R000237551.

1990 for example) to the conclusion that the best explanation for the structure of language is one which invokes biological evolution. In other words, the Language Acquisition Device is a biological adaptation that evolved through natural selection in response to the need to communicate "propositional structures over a serial interface" (Pinker and Bloom, 1990, 707).

2 Evolution without natural selection

The standard (biological) evolutionary approach to the origins of syntactic structure typically ignores the dynamics of the social/cultural transmission of language.[2] However, recently there has been more interest in the formal and computational properties of the influence of learning on the dynamic process of language transmission, historically, from one generation to the next (Niyogi and Berwick, 1997; Steels, 1997; Briscoe, 1998; Batali, 1998; Hurford, 1998). This paper explores the hypothesis that this process alone is enough to explain the emergence of the central unique features of syntax even where the prior bias of the learners is not subject to evolutionary change, and is relatively unconstraining.

Linguistic transmission can be viewed as a repeated transformation of information between two domains: the internal or I-domain, and the external or E-domain[3] (Kirby, 1999a). The I-domain contains languages that exist as grammatical knowledge in individuals' brains, whereas the E-domain contains languages that exist as sets of actual utterances. This conception has interesting parallels with the conception of genetic transmission in biology. Figure 1, compares the transformation of information between phenotypic and genotypic domains with the transformation between E- and I-domains.

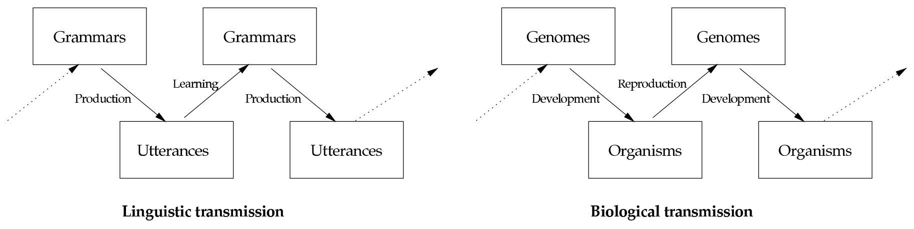

Fig. 1. Transformation of information in linguistic and biological adaptive systems.

In both these adaptive systems, the biological and the linguistic, the transformations that map between the two domains can be seen as "bottlenecks" on the

[2] See (Briscoe, 1998; Kirby and Hurford, 1997) for counter examples of papers which look at the combined effects of cultural transmission and biological evolution.
[3] These domains are parallel to Chomsky's (1986) I-language and E-language.

transmission of information over time through the whole system. For example, a particular piece of genetic information may not persist because the phenotype that it expresses may not survive long enough to reproduce. Similarly, a grammatical regularity may not persist because the utterances that it gives rise to cannot successfully reconstruct it through learning. Just as the bottleneck on transmission of genetic information has implications for the structure of the organisms that emerge (this is the basis of the neo-Darwinian synthesis, after all), we should expect the equivalent bottleneck on the transmission of grammars to impact on the eventual structure of languages.

3 Modelling linguistic transmission

In order to test how the learning bottleneck influences the structure of language, a working model of linguistic transmission must be created. The model consists of a simple population of agents who have a pre-specified "world" of concepts about which they can communicate, and a pre-specified set of symbols with which they can create utterances. The possible communicative behaviour of an agent is determined by that agent's internal grammar which is learnt solely through observation of the behaviour of other agents in the population. The population model is "generational" in that members of the population die and are replaced, but the initial state of the agents is always the same, the agents are not prejudiced or rewarded according to their behaviour in any way, and the only information that flows through the system is in the form of utterances — in other words, there is no natural selection or biological evolution in the simulation.

Semantics The semantic space in the model — the space from which meanings of utterances for the agents is chosen — is made up of combinations of simple atomic concepts. These concepts include, for example:

john heather
loves admires
believes

These atomic elements can be combined into simple predicate-argument propositions, which may have hierarchical structure. For example:

admires(heather,john)
believes(heather,admires(john,heather))

The full set of propositions is infinite in principle, because using predicates such as believes propositions can be limitlessly nested. In the simulations presented here, there are five "object" concepts (such as john, heather etc.), five "action" concepts (such as loves, admires etc.), and five "embedding" concepts (such as believes, knows etc.).

Utterances The utterances of the agents are linear concatenations of the 26 lower-case letters of the alphabet. There is no principled limit to the length of utterance, and the shortest utterance is 1 symbol long. Pairs of strings of symbols and propositions make up the E-domain objects in the simulation. So, if the agents happened to speak English, an E-domain object might look something like:

$$< \texttt{heatherlovesjohn}, \textsf{loves(heather,john)} >$$

Grammars The I-domain objects in the model are a simple form of definite-clause grammars (DCGs). Critically, this choice of formalism does not build-in compositionality or recursion. Consider an agent that could produce the string `heatherlovesjohn` meaning loves(heather,john). Here are two (of many, many possible) grammars that this agent may have internalised:

$S/\textsf{loves(heather,john)} \rightarrow \texttt{heatherlovesjohn}$

$S/p(x,y) \rightarrow N/x \quad V/p \quad N/y$
$V/\textsf{loves} \rightarrow \texttt{loves}$
$N/\textsf{heather} \rightarrow \texttt{heather}$
$N/\textsf{john} \rightarrow \texttt{john}$

The S symbol in these grammars is the start symbol for the grammar, and N and V are arbitrary node labels. The material after the slash on the non terminals is the semantic representation of that node. For left-hand side non-terminals in this formalism, this semantic representation may be a fully specified proposition, an atomic concept, or a partially specified proposition (i.e. one with variables replacing concepts). The right-hand side non-terminals' semantics may only be variables.

The grammar on the right is compositional. The meaning of a sentence is made up by combining two strings of type N with a string of type V, and the semantics for the sentence is built up from the semantics of these individual sub-strings using the variables x, p and y. The grammar on the left, however, is completely non-compositional. In no way is the meaning of the string a function of its parts.

3.1 Learning

Clearly, the central part of the simulation is the agents' ability to take E-domain objects (string/meaning pairs) and build I-domain objects (grammars). In fact, the agents are essentially induction machines designed to store pairs of strings and meanings, and find possible generalisations over these pairs (if such generalisations exist). The algorithm presented here has been designed specifically with simulation tasks in mind — it is extremely simple and efficient. Although no claims are made for its efficacy as a practical learning algorithm for DCGs in general, it does model in a simple way the dual processes of rote learning and search for generalisation that should be the core of any theory of language acquisition.

The algorithm is described in some detail in Kirby (1999b), and outlined informally[4] in figure 2. It works in two stages for each string/meaning pair. Firstly, the individual pair is incorporated into the learner's grammar by addition of the simplest possible rule that will generate that pair. For example, give the string/meaning pair:

$$< \texttt{heatherlovesjohn}, \textsf{loves(heather,john)} >$$

the following rule would be added to the grammar:

$$S/\textsf{loves(heather,john)} \rightarrow \texttt{heatherlovesjohn}$$

The second stage of learning (which takes place after incorporation of each rule) is a process of searching for possible generalisations (essentially subsumptions) over pairs of rules. For example, given the pair of rules:

$$S/\textsf{loves(heather,john)} \rightarrow \texttt{heatherlovesjohn}$$
$$S/\textsf{loves(heather,gavin)} \rightarrow \texttt{heatherlovesgavin}$$

the algorithm will replace these with a more general rule (via the "chunking" step g.3):

$$S/\textsf{loves(heather,}x\textsf{)} \rightarrow \texttt{heatherloves}\ N/x$$

The N category is a newly invented category, with its own rules added to the grammar:

$$N/\textsf{john} \rightarrow \texttt{john}$$
$$N/\textsf{gavin} \rightarrow \texttt{gavin}$$

3.2 Invention

Sometimes, the agents may wish to express meanings for which their grammar does not provide a way of saying. In fact, in the initial stages of the simulation, this will be very frequent, since the population will start with no language at all. The model therefore requires some way for creative language invention to occur. The simplest way to model this is for the agents to produce a string of random symbols whenever they encounter a meaning they do not know how to produce. In the simulations presented here, the strings vary between 1 and 3 symbols in length.[5]

[4] The chunking algorithm is not given here for lack of space (see Kirby, 1999b for details). The example in the text shows a typical application of chunking.

[5] The complete invention algorithm is discussed elsewhere (Kirby, 1999b). The only complication for implementation purposes is what should be done if the agent knows how to say almost all of a particular meaning. The algorithm in effect carries out a top-down production for the meaning as best it can given the grammar, and generates a random string at the point in the tree where the generation fails. This method ensures that invention does not ever introduce new structure. In other words, the invented string will never be more compositional than grammatical strings that already exist' in the agent's language.

```
**Induction algorithm:** Given a meaning $m$, a string $s$ and a grammar $g$:

i.1 parse $s$ using $g$, if the parse is successful, then return $g$.
i.2 form $g'$, the union of $g$ and $S/m \to s$.
i.3 apply a generalisation algorithm to $g'$:
    g.1 take a pair of rules $< r_1, r_2 >$ from $g'$.
    g.2 if there is a category label substitution $c$ to $c'$, that would make $r_1$ identical
        to $r_2$, then rewrite all $c$ in $g'$ with $c'$, go to g.5.
    g.3 if $r_1$ and $r_2$ could be made identical by "chunking" a substring on either or
        both their right hand sides into a new rule or rules, then create the new rules,
        and delete the old ones in $g'$, go to g.5.
    g.4 if $r_1$'s right hand side is a proper substring of $r_2$'s and $r_1$'s semantics is identical
        to either the top level predicate or one of the arguments of $r_2$'s semantics, then
        rewrite $r_2$ in $g'$ to refer to $r_1$, go to g.5.
    g.5 delete all duplicate rules in $g'$.
    g.6 if any rules in $g'$ have changed, go to g.1
i.4 return $g'$.
```

Fig. 2. Outline of the induction algorithm.

3.3 Simulation cycle

For the simulations reported here, the simplest possible population model is used. At any one time there are only two individuals: a learner and a speaker. The speaker is required to produce an utterance for a meaning chosen at random, either by using its grammar or by employing invention if the grammar cannot generate a string (if the speaker invents, then it uses its utterance as input to its own learning to ensure consistency). The learner is given the randomly chosen meaning and the speaker's string, and uses the pair as input to the induction algorithm. This process is repeated a fixed number of times, and then the speaker is removed, the learner becomes a speaker, and a new learner with an empty grammar is introduced.

4 Experiments

The graphs in figure 3 show the average behaviour of ten simulation runs starting with different random seeds. In all the runs, each speaker first tries to produce 50 randomly chosen degree-0 meanings (i.e. those with no embedding), then 50 randomly chosen degree-1 meanings (with one embedding) and finally 50 randomly chosen degree-2 meanings (with two embeddings). The graphs show the proportion of the meanings of each type that the speakers were able to produce without resorting to random invention. The other line on the graphs show a measure of size of the grammars (number of rules divided by 300 to fit it on the same plot).

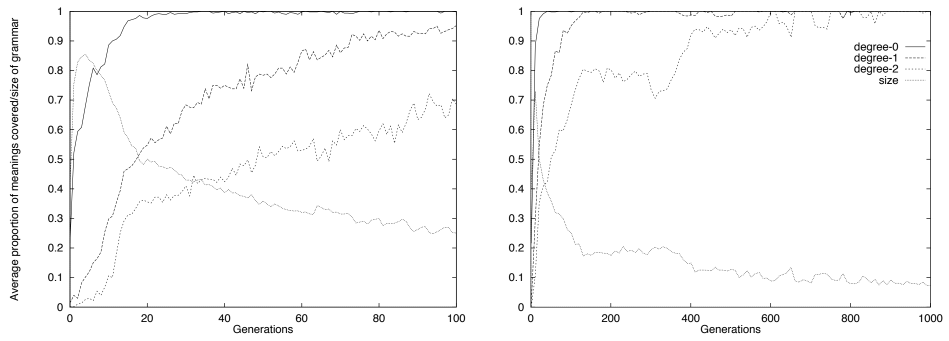

Fig. 3. Average of ten simulation runs, plotted on two different time-scales. Coverage of the different meaning types increases and the size of the grammar decreases over time.

To demonstrate more clearly what the behaviour of the evolving system is, we can examine the grammars of one population over time in some detail:

The emergence of vocabulary In the very early stages of the simulation, the speakers are not able to express many of the meanings without resorting to random invention — this is due to the fact that the population starts with no language at all. Immediately, there is a rapid rise in the size of grammars, as new innovated utterances are acquired by learners. The listing below shows part of an early grammar, which is best thought of as simply a long list of unrelated, idiosyncratic vocabulary items.

S/loves(pete,mary) $\rightarrow$ `axk`
S/thinks(john,likes(gavin,heather)) $\rightarrow$ `ew`
S/decides(heather,says(mary,detests(john,pete))) $\rightarrow$ `pq`
S/admires(pete,heather) $\rightarrow$ `njl`
S/decides(john,believes(pete,hates(pete,mary))) $\rightarrow$ `vz`
S/admires(gavin,mary) $\rightarrow$ `s`
S/knows(mary,decides(mary,loves(gavin,pete))) $\rightarrow$ `c`
S/detests(pete,john) $\rightarrow$ `my`
S/believes(pete,believes(mary,likes(mary,heather))) $\rightarrow$ `z`
S/likes(gavin,mary) $\rightarrow$ `but`
S/loves(gavin,pete) $\rightarrow$ `poe`
S/thinks(gavin,loves(heather,gavin)) $\rightarrow$ `oeb`
S/decides(mary,hates(gavin,john)) $\rightarrow$ `vri`
S/detests(heather,gavin) $\rightarrow$ `wb`
S/thinks(mary,hates(john,pete)) $\rightarrow$ `y`
S/thinks(john,hates(mary,pete)) $\rightarrow$ `pff`
S/decides(pete,detests(heather,mary)) $\rightarrow$ `fi`
S/believes(gavin,decides(gavin,admires(mary,john))) $\rightarrow$ `go`
S/believes(pete,knows(gavin,loves(heather,gavin))) $\rightarrow$ `jt`

.
.
.

The emergence of compositionality Figure 3 shows that after a brief initial rise the size of grammars soon starts to fall, and the proportion of degree-0 meanings that are expressed without invention increases. The grammar fragment below shows how this happens. Here, by the 100th generation, a compositional encoding has emerged, although it is only reliably used for degree-0 meanings. This particular grammar has two categories for arguments (arbitrarily named A and

C), which can be thought of as case-marked nominals, and one verbal category for predicates (named D). (N.B. there are many other idiosyncratic rules for more complex meanings in this grammar that are not shown.)

$S/p(x,y) \rightarrow$ `gj` C/y `z` A/x D/p
$A/$gavin $\rightarrow$ `dl`
$A/$heather $\rightarrow$ `tej`
$A/$john $\rightarrow$ `n`
$A/$mary $\rightarrow$ `qp`
$A/$pete $\rightarrow$ `h`
$C/$gavin $\rightarrow$ `x`
$C/$heather $\rightarrow$ `ovp`
$C/$john $\rightarrow$ `i`
$C/$mary $\rightarrow$ `h`
$C/$pete $\rightarrow$ `y`
$D/$admires $\rightarrow$ `b`
$D/$detests $\rightarrow$ `gl`
$D/$detests $\rightarrow$ `xe`
$D/$hates $\rightarrow$ `c`
$D/$likes $\rightarrow$ `e`
$D/$loves $\rightarrow$ `m`
$S/$knows(john,decides(pete,loves(mary,john))) $\rightarrow$ `qixoyia`
$S/$decides(x,thinks(heather,likes(pete,mary))) $\rightarrow$ C/x `jwyjtejdbznuy`
.
.
.

Here is an example sentences from this language with an English gloss (NOM and ACC stand for nominative and accusative):

(1) gj ovp z qp b
Heather-ACC Mary-NOM admires
"Mary admires Heather"

The emergence of recursion Finally, the percentage of degree-1 and degree-2 meanings that are expressible without invention increases as the size of the grammar continues to decrease. By generation 1000, an extremely concise language has emerged that can completely express the infinite meaning space (the entire grammar is shown below). This is done by using two verbal categories (B and D), one nominal category (A), and two sentence rules, one of which is recursive. In other words, a highly regular, expressive, syntactically structured language has emerged. This language persists without significant change for the rest of the simulation (which was terminated after 10000 generations).

$S/p(x,y) \rightarrow$ `gj` A/y `f` A/x B/p
$S/p(x,q) \rightarrow$ `i` A/x D/p S/q
$A/$gavin $\rightarrow$ `dl`
$A/$heather $\rightarrow$ `tej`
$A/$john $\rightarrow$ `n`
$A/$mary $\rightarrow$ `qp`
$A/$pete $\rightarrow$ `h`
$B/$admires $\rightarrow$ `b`
$B/$detests $\rightarrow$ `wp`
$B/$hates $\rightarrow$ `c`
$B/$likes $\rightarrow$ `e`
$B/$loves $\rightarrow$ `m`
$D/$believes $\rightarrow$ `g`
$D/$decides $\rightarrow$ `u`
$D/$knows $\rightarrow$ `ipr`
$D/$says $\rightarrow$ `p`
$D/$thinks $\rightarrow$ `m`

Here are examples from this language, again with glosses in English (notice how little the degree-0 structure has changed, apart from a simplification of the nominal system so that the nominative nouns are now used for accusatives as well):

(2) gj tej f qp b
Heather Mary admires
"Mary admires Heather"

(3) i h ipr gj tej f qp b
Pete knows Heather Mary admires
“Pete knows that Mary admires Heather”

(4) i dl p i h ipr gj tej f qp b
Gavin says Pete knows Heather Mary admires
“Gavin says that Pete knows that Mary admires Heather”

5 Conclusion: evolution through learning

The computational model shows how significant linguistic evolution can occur without any biological evolution. The learners in the simulation are not innately constrained to acquire only syntactic languages, nor are they rewarded in any way for their ability to communicate. The simulation is not seeded with any particular language — in fact, the first speaker has no linguistic knowledge to draw on whatsoever — and the only way information is transmitted over time is by observational learning. And yet, from this apparently simple set up, a linguistic system emerges. First, the language of the community is an impoverished vocabulary-like system, with idiosyncratic signals being assigned to whole complex meanings. Later, a compositional mapping emerges, with a lexicon for atomic concepts and a simple way of putting these together to form sentences. Finally, recursion evolves so that the agents are able, with only a small vocabulary and a couple of rules of combination, to express an infinite range of possible meanings.

The obvious questions that we are left with are “why does the language evolve in this way?” and “how general is this result?” Ultimately, a satisfactory answer to both these questions may well lie in a mathematical analysis of the properties of the transmission of information via observational learning. For the moment, however, it seems that the learning bottleneck that exists between the E-domain and I-domain in figure 1 must be what is driving the system to evolve.

Firstly, consider a non-compositional system such as the early ones in the simulation run described above. Here, the mapping between meanings and strings is such that each rule in the grammar can at best only account for one meaning-string pair. This necessarily means that to learn any particular rule in such a grammar, a learner must be exposed to exactly that pairing. The chance that a particular non-compositional I-language will survive over time are slim in this situation, because every string-meaning pairing in that I-language will have to appear in the input data to the learner. Given that the number of utterances that the learner hears is smaller than the number of meanings that a speaker might be called upon to produce, then a randomly chosen, complete, non-compositional language *cannot* be transmitted through the learning bottleneck. Notice, this must be true for any learning algorithm.

Now, contrast this situation with one in which there is a compositionality in the language. Given an E-language in which there is some pattern to the string-meaning mapping, it is no longer true that a particular pairing can only persist if it is directly observed. A learner can generalise from an observed set of string-meaning pairs to ones that are unseen. This means that a compositional language will be far more transmittable through the bottleneck. A smaller set of example pairings will required in order to reconstruct the language accurately. Again, as long as the learning algorithm is able to exploit pattern, or is biased towards generalisation, then this is true in general.

It seems as if the process of the transmission of learned behaviour has certain general properties whatever our assumptions about the learners. These properties will tend to favour the emergence of systems with patterns that can be exploited by learning. Where the behaviour that is being learned is a mapping between two structured spaces, we can expect the process of transmission to give us mappings that preserve structure. For the case of language — where these spaces are propositional meanings on the one hand, and linear strings on the other — the mappings that will inevitably emerge will be syntactic.

References

Batali, J. (1998). Computational simulations of the emergence of grammar. In J. Hurford, C. Knight, and M. Studdert-Kennedy (Eds.), *Approaches to the Evolution of Language: Social and Cognitive Bases*, Cambridge, pp. 405–426. Cambridge University Press.

Briscoe, E. J. (1998). Language as a complex adaptive system: co-evolution of language and of the language acquisition device. In P. Coppen, H. van Halteren, and L. Teunissen (Eds.), *8th Meeting of Comp. Linguistics in the Netherlands*, Amsterdam, pp. 3–40. Rodopi.

Chomsky, N. (1986). *Knowledge of Language.* Praeger.

Hurford, J. (1998). Social transmission favours linguistic generalisation. In C. Knight, J. Hurford, and M. Studdert-Kennedy (Eds.), *The Emergence of Language.* To appear.

Kirby, S. (1999a). *Function, Selection and Innateness: the Emergence of Language Universals.* Oxford: Oxford University Press.

Kirby, S. (1999b). Learning, bottlenecks, and the evolution of recursive syntax. In E. J. Briscoe (Ed.), *Linguistic evolution through language acquisition: formal and computational models.* Cambridge: Cambridge University Press. Forthcoming.

Kirby, S. and J. Hurford (1997). Learning, culture and evolution in the origin of linguistic constraints. In *Fourth European Conference on Artificial Life*, pp. 493–502. MIT Press.

Niyogi, P. and R. Berwick (1997). Populations of learners: the case of Portugese. Unpublished manuscript, MIT.

Oliphant, M. (1998). Rethinking the language bottleneck: Why don't animals learn to communicate? In C. Knight, J. Hurford, and M. Studdert-Kennedy (Eds.), *The Emergence of Language.* To appear.

Pinker, S. and P. Bloom (1990). Natural language and natural selection. *Behavioral and Brain Sciences 13*, 707–784.

Steels, L. (1997). The synthetic modelling of language origins. *Evolution of Communication 1*(1).