Language Acquisition and Implications for Language Change: A Computational Model*

Robert A. J. Clark

Department of Linguistics,
University of Edinburgh
robert@cstr.ed.ac.uk

Abstract

Computer modeling techniques, when applied to language acquisition problems, give an often unrealized insight into the diachronic change that occurs in language over successive generations. This paper shows that using assumptions about language acquisition to model successive generations of learners in a computer simulation, can have a drastic effect on the long term changes that occur in a language. More importantly, it shows that slight changes in the acquisition model can have drastic effects on language change.

1. Introduction

This paper looks at the issues of language acquisition and language change by taking the simple three parameter grammar discussed by Gibson & Wexler (1994) and evaluating it in the context of a model proposed by Niyogi & Berwick (1996), based on the Trigger Learning Algorithm (Gibson & Wexler 1994)

The three parameter grammar is based on a two parameter grammar which defines base word order (i.e the underlying word order in an utterance) with an additional parameter which causes verb movement when set to give a surface structure which differs from the original base structure.

The first two parameters define the base word order. We take the the following X-bar schemata:

$$(X1) \qquad X' \rightarrow X, Complement,$$

$$(X2) \qquad XP \rightarrow X', Specifier,$$

where rule $(X1)$ states that X(head) and $Complement$ are sisters, but says nothing about ordering. Similarly, rule $(X2)$ states that X' and $Specifier$ are sisters. Two binary-valued parameters are then employed to set the ordering. The first parameter *spec-first* when set to 1 defines a specifier-first language and when set to 0, a specifier-final language. Similarly the second parameter *comp-first* defines a complement-first language when set to 1 and a complement-final language when set to 0.

A third parameter is now introduced. When set, this parameter forces:

> a finite verb to move from its base position to the second position in root declarative clauses. (Gibson & Wexler 1994)

This means that although the underlying base word order is unchanged, the surface structure may be reordered. Theories about why verb movement occurs are varied, (see Gibson & Wexler (1994) for more details), but the mechanism assumed here is that the verb moves to the C position.

The grammar space defined by these parameters consists of 8 grammars made from 4 base word orders, each with or without verb movement. These grammars can be referred to by their base word ordering, followed by either $+V2$ or $-V2$ to signify the presence of verb movement. Table 1 lists each parameter setting along with its given name.

spec	*comp*	*V2*	resulting grammar
0	0	0	VOS -V2
0	0	1	VOS +V2
0	1	0	OVS -V2
0	1	1	OVS +V2
1	0	0	SVO -V2
1	0	1	SVO +V2
1	1	0	SOV -V2
1	1	1	SOV +V2

Table 1: The parameters and their resulting grammars

Gibson & Wexler imply that it is generally accepted in the (psycho)linguistic literature that a learning algorithm which relies on the existence of triggers is assumed. They put forward the Trigger Learning Algorithm (TLA) as a natural choice:

> Given a set of initial values for n binary valued parameters, the learner attempts to syntactically analyse an incoming sentence S. If S can be successfully analysed, then the learner's hypothesis regarding the target grammar is left unchanged. If, however the learner cannot analyse S, then the learner uniformly selects a parameter P (with probability $1/n$ for each parameter), changes the value associated with P, and

*This research was completed in partial fulfillment of the requirements for an MSc by the University of Edinburgh (Clark 1996), and was supported by a studentship from EPSRC.

tries to re-process S using the new parameter value. If analysis is now possible, then the parameter value change is adopted. Otherwise the original parameter value is retained. (Gibson & Wexler 1994)

Figure 1 shows the steps of a single iteration of the algorithm diagrammatically. It shows the result of processing a single utterance with respect to the current hypothesis.

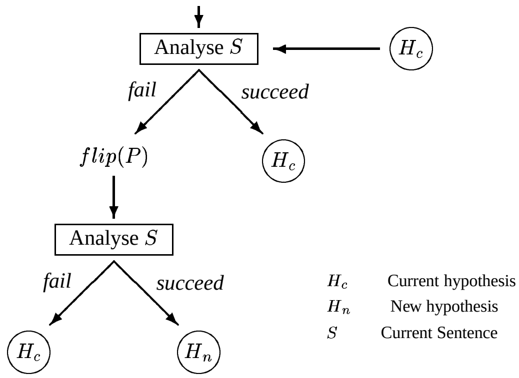

Figure 1: The Trigger Learning Algorithm

The TLA is based around two specific constraints: The first constraint makes the TLA conservative, in that it only considers grammars that differ from the current hypothesis by one parameter, and is known as the Single Value Constraint (Clark 1990). The second constraint requires the new hypothesis to give an analysis of the current sentence and is referred to as the Greediness Constraint (Clark 1990). The algorithm is greedy in the sense that it is only prepared to change its current hypothesis if it can gain something from it (namely an analysis for the current sentence). Gibson & Wexler prove that this algorithm with the above constraints mathematically guarantees that a learner can converge to a target grammar in the limit (i.e. after sufficient iterations) with input that consists of positive examples from this grammar when every grammatical hypothesis is associated with a local trigger.

We consider two formulations of the model. Initially we take the model in it basic form. We then try to resolve some of the problems encountered by adding a parsing complexity measure.

2. Evaluating the Basic Model

To evaluate this model we implement it as a computer simulation. We use transition matrices representing the Markov process modeling acquisition, and probabilistic distributions representing the population. This models the theoretical outcome of each step in the model in terms of proportions, instead of individually and randomly simulating a large number of learners hearing individual utterances. This method saves computational time, as it is effectively only modeling one *average* learner. It models the gross behavior of the system, without the need to model a large number of individual learners. In addition, this method removes the need to run a particular model many times to produce statistically significant results, as the method produces the statistical outcome directly. The basic procedure is as follows:

1. Take the parameters being used and define the resulting languages.

2. Set up the distribution for the initial population of speakers: Here each language (or parameter configuration) is assigned a value [0–1] which represents the proportion of the population which speaks that language. The main sets of initial conditions that were experimented with included homogeneous populations of each language, a mixed population speaking an equal proportion of each language and a mixed population speaking each of the -V2 languages.

3. Set up the primary linguistic data: Knowing the linguistic composition of the population, (from the previous step) each utterance that is producible within the defined grammars is allocated a value [0–1] to represent the probability that it could be the next utterance produced by that population.

4. Calculate the transition matrix: The transition matrix represents the Markov process modeling acquisition. The effect that hearing an utterance has on a learner is stored in the matrix as the probability of shifting from one hypothesis to another.

5. Calculate product matrix: The transition matrix is then raised to the power m (multiplied by itself m times), where m is the number of utterances after which a learner matures. This matrix then provides the information to show what proportion of a population of learners will acquire each language in the language space.

6. Generate the new population of speakers: The product matrix is then used to generate a new population of speakers for the next generation.

7. Return to Step 3

Transition matrix entries of non-neighbouring states are zero, otherwise entries are calculated by the following:

$$\begin{aligned} P[s \rightarrow k] &= \sum_{s_j \notin L_s, s_j \in L_k} \frac{1}{n} P(s_j), \\ P[s \rightarrow s] &= 1 - \sum_{\substack{k \text{ a neighbouring} \\ \text{state of } s}} P[s \rightarrow k]. \end{aligned} \tag{2}$$

L_s and L_k are the languages defined by the grammars s and k respectively. The transition from s to k is made up of a components relating to each utterance's frequency in the Primary Linguistic Data (PLD), that is, the set of possible utterance that can be spoken by the population, $(P(s_j))$.

3. The Outcome of the Basic Model

The behaviour of the model depends on three factors:

1. Maturation time of learners.

2. The initial hypothesis of a learner.

3. The input data given to learners (i.e. the utterances they process)

Maturation time affects language change quite considerably. A very low maturation time (i.e. after only two or three utterances) results in approximately equal proportions of all languages being spoken in the next generation, with only a slight bias from the distribution of speakers in the current population. (This is what would intuitively be expected: a learner's initial hypothesis is chosen at random, and a low maturation time allows very little data to be presented to support or oppose this hypothesis. No time is allowed for convergence to a particular grammar to occur, thus the final hypothesis is close or identical to the initial randomly selected hypothesis.) For the evaluation of effects on language change the maturation time is chosen to allow for reasonable convergence.

Initial parameter settings may also affect convergence, especially with a low maturation level. Initial parameter settings for learners are random in the formulation of the model discussed here.

The input data also effects the behaviour of the model. The model assumes that the utterances produced by a speaker of a particular language are uniformly distributed across the utterances which are part of that language. That is if a and b are both constructions of language L, that a speaker of L is equally likely to produce the construct a as he is the construct b. It is unlikely that in a real language all the constructions are used to the same extent. However, for simplicity, in the model discussed here all productions are assumed equally.

The results this model produces fall into three basic categories.

1. S-shaped logistic change. This type of change, discussed by Kroch (1989) and Kirby (1996), is the behaviour Niyogi & Berwick show the model to produce. Regarded as the pattern of change found historically in linguistics and more widely in biological systems of change.

2. No Change. There are situations, specifically with some initial homogeneous populations, where no change is seen from one generation to the next.

3. Other behaviour. There are more general cases where language change occurs in a fashion not corresponding to either of the above categories, usually in the form of exponential change.

Running the model with a wide range of initial conditions reveals that these particular parameters favour +V2 languages. The +V2 languages in this parameterisation have more constructions than –V2 languages and the organisation of the utterances is such that it is easier to set the verb movement parameter than it is to unset it, that is on average there are more triggers to set the verb-movement parameter than to unset it. An example of a –V2 language being taken over by a +V2 language is shown in Figure 2. This phenomenon could be a direct result of the parameters being used and the fact these parameters do not reflect the real world, or that other factors not accounted for by the model, such as the distribution of utterances which are produced within any one language, meaning that this model alone is too simple to reflect the real world. We should be aware of this when interpreting results.

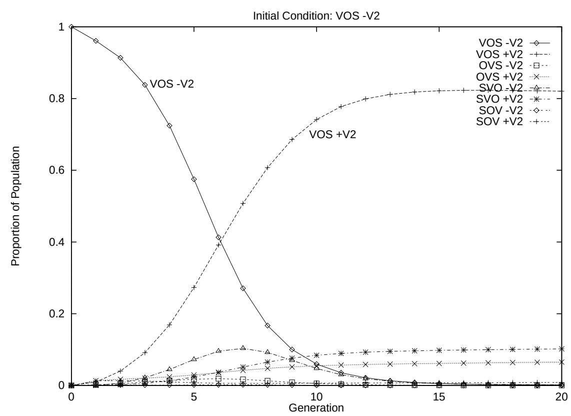

Figure 2: Graph showing logistic change. Initial population speaking VOS –V2, 80% of final population speaking VOS +V2

We also see that the language change that occurs generally happens exponentially and not with the S-shape behaviour which Niyogi & Berwick suggest. These issues are the motivation for the modifications which we make to the model.

4. The TLA with a Complexity Factor

To examine this model more closely, specifically with respect to its stability of outcome, and to look more closely at the issue of +V2 languages dominating, we introduce ideas to account for parsing complexity into the model, whilst attempting to stay within the spirit of the TLA.

To do this we alter the algorithm as shown in Figure 3. Here the left failure branch of the algorithm remains unchanged and a new hypothesis is searched for if the parse fails. However, if the parse succeeds, the current hypothesis grammar is rejected for a fitter hypothesis grammar if one can be found by randomly flipping a single parameter. The transition matrix calculation is formalised in Equation 3.

$$
\begin{aligned}
P[s \rightarrow k] &= \frac{1}{n}\left(\sum_{\substack{s_j \in L_s,\, s_j \in L_k \\ F_s(s_j) > F_k(s_j)}} P(s_j) + \sum_{s_j \notin L_s,\, s_j \in L_k} P(s_j)\right), \\
P[s \rightarrow s] &= 1 - \sum_{\substack{k \text{ a neighbouring} \\ \text{state of } s}} P[s \rightarrow k].
\end{aligned}
\tag{3}
$$

It should be apparent that for this formalism to reject an acceptable current hypotheses, utterances must have different complexity values with respect to different grammars. If the complexity of an utterance is independent of the grammar that produced it, and hence is the same across all grammars, then an acceptable current hypothesis will never be rejected on the grounds that no other acceptable hypothesis is any better than the current one.

The verb movement to second position in the surface structure of an utterance is an obvious candidate to use to measure complexity, where complexity values for the same utterance must differ with respect to different grammars. To demonstrate the above formalism the following example metric can be used:

$$
F_k(\omega) = \begin{cases} 1 & \text{if } k \text{ is a } +V2 \text{ language} \quad \forall \omega \in F_k \\ 0 & \text{if } k \text{ is a } -V2 \text{ language} \quad \forall \omega \in F_k \end{cases}
$$

This ranks all utterances which are parsed with verb movement as being more complex than those which are parsed without. That is, any utterance parsed by a $+V2$ grammar is complex, and any utterance parsed by a $-V2$ grammar is simple (e.g. the construction SVO is complex when parsed by a $+V2$ grammar and simple when parsed by a $-V2$ grammar, and the utterance VSO is always considered simple as it only occurs in $-V2$ languages).

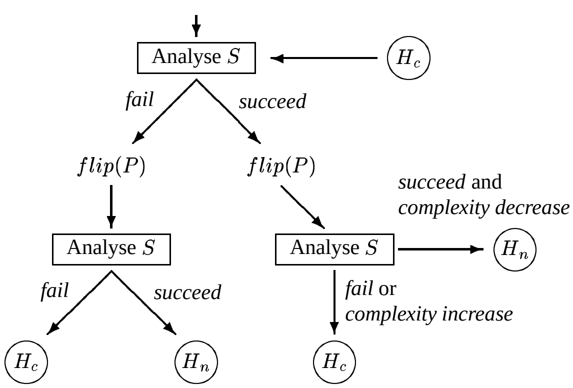

Figure 3: The Trigger Learning Algorithm with a Complexity Factor

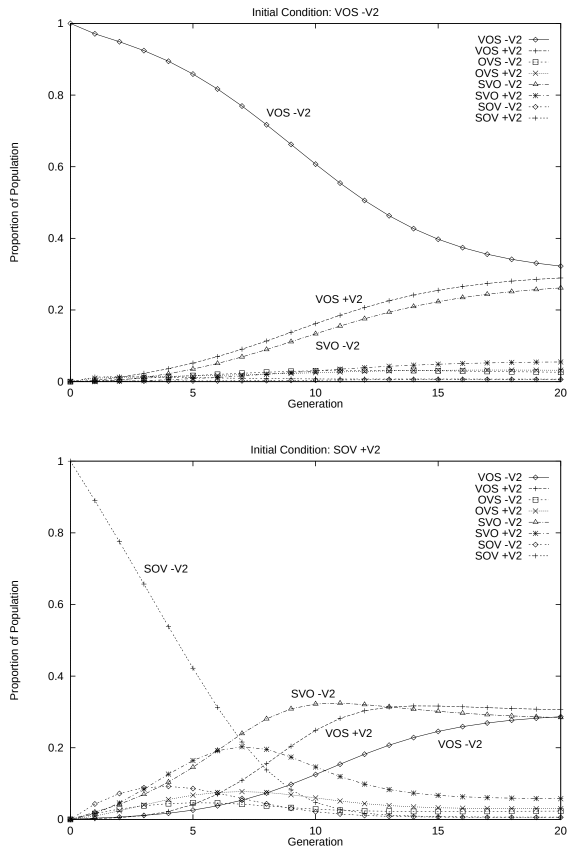

Figure 4: Graphs showing outcome with complexity factor.

Examples of the outcome of the altered TLA with this metric are shown in Figure 4 and the behaviour can be summarised as follows:

1. The same steady state is reached after about 30 generations, in that each of the languages SVO $-V2$, VOS $-V2$ and VOS $+V2$ end up spoken by approximately 30% of the population and all other languages are spoken by less than 7% of the population, none of them being eliminated completely. The fact that languages can survive in very small proportions and not be completely wiped out is of interest because there exist in the world language types which survive in very small proportions. This result is in contrast to that seen with the basic model, where language survival is often all or nothing.

2. Intuitively it might be expected that by marking all $+V2$ utterances as complex, effectively

making them less fit, a strong advantage is being given to $-V2$ languages and they would dominate over $+V2$ languages. However, this is not what the results show. Out of the three languages that dominate one of them is a $+V2$ language.

3. Logistic change is now much more widespread, possibly reflecting the competition between rival forms that is present. If this is how real language change progresses over time, then it is possible that parsing complexity, or some similar filtering mechanism, should be an issue in any such model of language change.

4. There is only a slight internal change to the model. The result of adding V2 complexity factor to the algorithm is that some of the internal transitions in the model that are from a +V2 state to itself, are altered to being from the +V2 state to the –V2 state. In total there are 15 such altered transitions. Initially 250 of the 576 transitions (1 from each state for each utterance) result in the model changing state. So the change from the basic formulation of the model is caused by increasing the number of set transitions (i.e. not from a state to itself) by 15 (6%), which is equivalent to altering 2.6% of the total number of transitions.

With respect to the steady state reached, it is difficult to compare this result to historic change, as issues such as growing population sizes and spatial distributions are not considered by the model. The important difference between the basic model and the model proposed here though, is that depending on the initial conditions the basic model reaches different steady states, whereas the extended model always reaches the same steady state.

5. Conclusions

It is probably wrong to draw too many conclusions from either of these models about real language. The models are rather simplistic and can easily be criticised for their failing to represent real language scenarios.

Additionally, there are still a number of areas that need further work and many issues that are still unclear. The V2 complexity model proposed here only assigns binary values (1 and 0), splitting the grammars into two subsets: one composed of languages of fit utterances and one composed of languages of unfit utterances. A metric over the whole interval would reflect current theories (e.g. those of Hawkins (1994) and Kirby (1996)) more accurately. There is also no interaction within the current metric; that is, fitness is the result of only one factor. How interaction in the metric would effect the model, (c.f. the way surface word order is the result of the interaction between base word order and verb movement) needs to be more closely examined. It would also be interesting to know to exactly what extent the behaviour of the model depends on a particular complexity metric employed and the parameterisation of the grammars. Is the resulting stable solution dependent upon the parameter interaction or upon the parsing complexity? Could other weak forms (like the complex +V2 language that is found to survive), survive under different parameterisations or different complexity metrics?

What is clear, however, are the drastic effects that small changes to the models' formalism can have on the outcome, specifically in the context of language change. Smooth logistic change is more widespread with the altered TLA. This suggests that the model may reflect historical change better than in its unaltered form. This in turn suggests that both the effects of parsing complexity and underlying parameterisation need to be accounted for by a model; and that a model which concentrates on a purely parameter based approach or a purely functionalist approach is insufficient.

The main point to be made though is that language acquisition has big implications for language change. Small differences in each generation in a model of acquisition soon accumulate and drastically affect the pattern of language change.

References

Clark, R. (1990), *Papers on Learnability and Natural Selection*, Technical Reports in Computational Linguistics, No. 1.

Clark, R. A. J. (1996), Internal and external factors affecting language change: A computational model, Master's thesis, Dept. of Linguistics, University of Edinburgh.

Gibson, E. & Wexler, K. (1994), 'Triggers', *Linguistic Inquiry* **25**(3), 407–454.

Hawkins, J. A. (1994), *A Performance Theory of Order and Constituency*, Cambridge Studies in Linguistics, Cambridge, UK.: Cambridge University Press.

Kirby, S. (1996), Function, Selection and Innateness The Emergence of Language Universals, PhD thesis, Department of Linguistics, University of Edinburgh.

Kroch, A. S. (1989), Function and grammar in the history of English: Periphrastic "do.", *in* R. Fasold, ed., 'Language Change and Variation', Amsterdam: Benjamins.

Niyogi, P. & Berwick, R. C. (1996), 'A language learning model for finite parameter spaces', *Cognition* **61**, 166–193.