In the Proceedings of the 28th annual Boston University Conference on Language Development, 2004.

Evaluating Models of Parameter Setting

Janet Dean Fodor[1] and William Gregory Sakas[1,2]
[1]Graduate Center, City University of New York
[2]Hunter College, City University of New York

1. The research project

Our research group has built a research environment for testing models of syntactic parameter setting. We have created a domain of 3,072 languages with up to 1,432 sentences in each, defined by a set of Universal Grammar (UG) rules and 13 (so far) binary parameters. The languages resemble human languages in many respects, though they are considerably simpler (details below). Each of these languages is designated in turn as the target for acquisition and its sentences are input to a learning algorithm. We measure whether learning is successful (is the target reliably attained?) and how long it takes (how many input sentences the learner consumes before arriving at the target grammar). Our goal in preparing this rich testing environment was to further the search for a credible psycho-computational model of syntactic parameter setting. That sounds very grand, but what it means is just: a model that is psychologically realistic, precisely specified, compatible with linguistic principles, and as reliably successful as children are. This has proven remarkably elusive.

Individual grammars are defined by their parameter values, so acquiring a language consists in identifying the parameter values that license it.[1] When the principles and parameters (P&P) theory of grammars was first proposed (Chomsky, 1981), it was hailed as a sweeping solution to problems that had

* This work is presented on behalf of the CUNY Computational Language Acquisition Group (CUNY-CoLAG), and was supported by grants from The City University of New York PSC-CUNY Research Award Program. Much credit is due to colleagues and students who have contributed to this research: Carrie Crowther, Atsu Inoue, Xuan-Nga Kam, Yukiko Koizumi, Eiji Nishimoto, Artur Niyazov, Yana Pugach, Lisa Reisig-Ferrazzano, Iglika Stoyneshka-Raleva, Virginia Teller, Lidiya Tornyova, Erika Troseth, Tanya Viger, Sam Wagner. The language domain described below, and other information, is available at www.colag.cs.hunter.cuny.edu.

[1] When we refer to a grammar in what follows we will mean the syntactic component of a grammar (the computational system). The lexicon must also be learned, but we have set that aside in our current research while recognizing the significant problems posed by the interaction of syntax and lexicon. We also do not consider the acquisition of phonology or morphology, and we assume universal semantics.

been building up within traditional attempts to model language acquisition as rule creation. If the whole grammar is innate except for just 20 (or even 200) binary parameters of variation, then a child has only 20 (or 200) simple facts to acquire from the input. The child's task is focused by the parametric format. It demands no creativity and only minimal data processing. The mechanism for assigning a value to a parameter is called 'triggering', a process held to be fast, accurate and 'automatic', the latter implying that it requires no linguistic computation on the part of the learner. However, the triggering process has never been precisely modeled in psycholinguistics or computational linguistics. It seems clear that it never will be, for reasons that are now well understood. In important work through the 1990s, Robin Clark, Ted Gibson, Kenneth Wexler and others have shown that parameter setting cannot be labor-free and is not always successful. Parameter setting as a concept is still paramount, but parameter setting as a process has become a source of problems rather than solutions.

Several models of the parameter setting process have been proposed in the literature. A striking fact is how little they resemble the pre-theoretic notion of triggering. Acquisition is viewed, rather, as a search through the field of all possible grammars to find one that licenses the sentences in the input sample. The ingenuity in these models consists in their adaptation to language of various general-purpose search techniques for large-scale domains. In our own work we have tried instead to stay as close as possible to the original notion of triggering, retaining the central aspect that an input sentence guides the learner toward the particular parameter(s) that need to be reset. We believe this is psychologically more plausible than the grammar search techniques, and also that it is more efficient. But that needs to be shown. It is extremely difficult to gauge the efficiency, or even the reliability, of a proposed learning procedure from a description of it in the abstract. Language structure is so intricate, and the number of potential target languages a learner must be equally capable of acquiring is so great (even for a relatively small number of parameters), that it not easy to anticipate how a particular learning algorithm will perform when actually put to the test. Simulation studies are therefore essential.[2]

Previous simulation studies have varied in scope. Some evaluate a single learning model in a miniature natural-language-like domain. Most notably, Gibson & Wexler (1994) tested their Triggering Learning Algorithm (TLA) on a 3-parameter domain parameterized for word order. This domain was subsequently expanded. Bertolo et al. (1997) added verb raising, and word order parameters for embedded clauses. Kohl (1999) studied Bertolo's full 12-parameter domain which included also several parameters for scrambling. Clark (1992) and Nyberg (1992) have employed a large but abstractly characterized

[2] Mathematical methods, as in the tradition of Gold (1967) and Angluin (1980), have yielded important general results (see Jain et al., 1999, for discussion and references), but they require simplifying and homogenizing assumptions somewhat far removed from the realities of human language.

domain (30 parameters) in which the 'sentences' consist of codings of their licensing parameter values. Briscoe (2000) and Villavicencio (2000) have employed categorial-grammar-based domains (up to 28 hierarchically organized parameters generating up to 800 languages). Villavicencio used one target language (English) as the test-case for her learner; the input was based on a transcript of child-directed speech. Yang (2002) has tested sets of two or three competing parameters, drawn from a larger domain, against statistics derived from the CHILDES database (MacWhinney 2000). What our simulation project adds to this earlier work is the ability to compare the performance of a range of different learning models, all exposed to the same large collection of realistically-structured languages. The results we report here are preliminary. There are many more empirical questions to be asked and answered about these models. But among those we have tested we are finding the best combinations of efficiency and psychological fidelity among our own class of learning models, the *Structural Triggers Learners* (Fodor 1998; Sakas & Fodor 2000; Sakas & Nishimoto 2002).

2. Parameter setting is difficult

Why has parameter setting been hard to model? The trouble, in a nutshell, is that sentences do not always announce which parameter values license them. A learner's task is to determine which parameter values were employed by the speaker who uttered the sentence, but the learner may not even be able to tell which values *could* have yielded that sentence. In consequence, even a sentence which is in fact a fully unambiguous trigger for a particular parameter value may convey very little information to the learner. The causes of this can be grouped into two kinds: parameter interaction problems and parametric ambiguity problems.

2.1 Parameter interaction problems

Clark (1988, 1992) has drawn attention to the fact that even if parameters are formally independent, they interact with each other in derivations. The surface string, which is what a learner is exposed to[3], reflects the combined effects of UG principles and all the parameter values that contributed to the derivation of the sentence. As a result, a given parameter value may have no distinctive isolatable effect on sentences. That is: it has no trigger, in the strong sense of the term. (In a cue-based learning model such as proposed by Lightfoot, 1991, for syntax, and Dresher, 1999, for phonology: it has no cue.) Learners

[3] As emphasized by Pinker (1984), Morgan et al. (1987) and others, the semantic, prosodic and morphological properties of sentences can provide learners with cues to the syntactic structure of their input, though these properties were traditionally disregarded in computational learning research. We are just beginning to incorporate them into our language domain.

must therefore engage in what we have called *parametric decoding*. They must somehow disentangle the derivational interactions in order to identify which parameter values a sentence requires. Unlike the older conception of 'instant triggering', this parametric decoding evidently cannot be automatic or effortless. It is hard work precisely because parameters work in concert with each other. How it is done is what distinguishes one learning model from another.[4]

Parameter values are not transparently displayed by input sentences, so learners can't just read them off. It seems, then, that a process of trial and error is inevitable. The learner must try out some possible parameter values and see whether they work. Since the effects of individual parameters cannot always be isolated from each other, it also seems inevitable that parameter values must be tried out in combination rather than singly. This explains why recent research has tended toward models in which a whole grammar is selected for testing against an input sentence, to see whether or not it is compatible with it, i.e., whether it can parse it. Trial and error on whole grammars is about as different from triggering as can be imagined, but it does squarely confront the parameter interaction problem. On the other hand, it does not obviously recommend itself in terms of efficiency. It can be implemented in various ways, but none is ideal. If a single grammar is tested per input sentence, many learning opportunities will be wasted due to unlucky guesses as to which grammar was worth trying. If many grammars are tested per input sentence, the hit rate will be higher but the computational load will exceed plausible psychological limits. The number of candidate grammars is huge (see below), and in the worst case every one of them must be tested in order to identify a single correct parameter value. Suppose value v_i of parameter *P* is necessary to license a particular target sentence, but that it does so only if all other parameters are correctly set. An unlucky learner might try out each value of *P*, in the company of all combinations of values of the other parameters, and fail consistently before eventually hitting on the right combination.

2.2 Parametric ambiguity problems

Parametric ambiguity occurs when a sentence (a surface string) belongs to two or more possible languages, generated by distinct grammars (= distinct collections of parameter values). A parametrically ambiguous sentence does not reveal which parameter values licensed it. It would not do so even if the decoding problem had been solved. At best, decoding could indicate the range of *candidate* parameter value combinations. Then the learner could guess among those, and not waste effort on grammars that couldn't possibly be the target. Alternatively, when decoding reveals ambiguity, a learner might discard that

[4] For models not assuming syntactic parameters, or any UG at all, the same problem arises but in a different guise. The learner must still, in some fashion, establish what an input sentence reveals about the target grammar and this may be opaque.

sentence in order to avoid the risks of guessing. However, this 'waiting' strategy places even heavier demands on decoding than a guessing strategy does, since it assumes that the learner can know when *more* than one grammar licenses the sentence, so its search for candidates must be exhaustive. In fact, the difficulties of decoding are so great that most current models do not even aspire to ambiguity detection (multiple decoding). However, a Structural Triggers Learner (STL) is capable of detecting and responding appropriately to ambiguity, due to its relatively efficient decoding process, as we explain below.

How much parametric ambiguity there is in natural language has never been quantified, but it is surely considerable. The question is: How much overlap is there between the sentences of distinct languages? Assuming that children receive (or attend to) relatively simple sentences, the amount of ambiguity they face is likely to be especially high. There are not that many distinct ways of arranging a few basic sentence constituents such as subject, verb and object, so any one arrangement is likely to occur in many languages. More complex sentences are typically less ambiguous, but also less accessible at early stages of learning. The amount of cross-language overlap (for a given number of potential sentence patterns) depends on how densely packed the language domain is, and this is a function (in part) of how many languages there are in the domain. For the natural language domain, that number is high on all current estimates. We discuss the matter of scale in section 2.4 below. The point we emphasize here is that more languages can mean not just a larger search space to scour for the target, but also more ambiguity, i.e., less information per input sentence.

In our artificial domain, the most ambiguous sentence (which consists of a single verb) occurs in all 3,072 languages; the least ambiguous sentences occur in just two languages; and the mean for all 28,924 sentences in the domain is 326 languages.[5] Thus, even with complete and perfect decoding, the average probability of guessing correctly by pure chance would be only 1 in 326. Of course there is no reason to suppose that these proportions are characteristic of the whole domain of natural languages (especially as ambiguity is inflated in our domain by the simplicity of its sentences), but this does give some sense of how severe the ambiguity problem can be.

2.3 Examples from the CUNY language domain

[5] These numbers include parametric irrelevance folded in with parametric ambiguity. In fact, in every case where a sentence sets all but one parameter, that parameter is irrelevant to the sentence (e.g., the Pied Piping parameter for a sentence with no prepositional phrases). In other words, the domain includes some fully unambiguous triggers. On the other hand, this measure of ambiguity gives a low estimate of the amount of parametric uncertainty for learners, because a sentence may have several derivations using different parameter values even in the same language.

Examples (1)-(3), from our language domain, illustrate both parameter interaction and parametric ambiguity. Readers may enjoy the challenge of decoding these sentences, though we caution that the exercise may be frustrating because the parameter names are only a rough guide to what the parameters actually do, and we lack space to specify them in detail here. These 'sentences' look odd because we abstract away from lexical variation across languages by using grammatical category symbols as a universal vocabulary for all languages in the domain (a strategy borrowed from Gibson & Wexler, 1994). The structures of these sentences are well-defined. They are generated by a universal grammar supplemented by the parameter values. There are 13 binary parameters (not all fully independent; see Table 1), which participate in familiar P&P (government-binding theory) analyses. For ease of implementation, however, the grammar is represented in a generalized context-free phrase structure format. The parameter values take the form of underspecified PS rules, or subtrees ('treelets'). For example, the parameter value that licenses Wh-Movement is a treelet consisting of a Spec,CP with a [+WH] feature. Any language which has this parameter value treelet can make use of it in generating sentences. UG entails that a Spec,CP[+WH] will always dominate a constituent that is marked[+WH] and is associated (via SLASH features) with a trace created elsewhere in the sentence. (Note: O1 is a direct object. O3 is the object of a preposition. *KA* is a question marker, borrowed from Japanese, which appears in all interrogative sentences in all languages if the finite verb has not raised to C^{o}.)

(1) **O3 Verb Subj O1[+WH] P Adv.**
This sets: no wh-movement, preposition-stranding, head initial VP, V-to-I movement and I-to-C movement, no affix hopping, C-initial, subject initial, no overt topic marking. It is ambiguous with respect to: obligatory topic, null subject, null topic.

(2) **Adv[+WH] P NOT Verb Subj KA.**
This sets all parameters except ±overt topic marking. For example, it sets null topic and no null subject because the absence of an overt O3 can only be due to topicalization of O3 followed by topic-drop (i.e., null topic), and UG specifies that null topic is mutually exclusive with null subject.

(3) **O1-WA Verb.**
This sets +overt topic marking and +null subject, which entails –null topic and –obligatory topic. (Without -WA, this sentence would set no parameters at all.)

2.4 Problems of scale

The effort of parametric decoding and the degree of parametric ambiguity can both be exacerbated by the size of the domain. How large is the natural language domain? If there are *n* independent binary parameters, there are 2^n

possible languages. So now we need to know how many natural language parameters there are. Linguistic research aims to keep their number low, by showing that several sources of language variation are correlated and so can be attributed to a single parameter. Nevertheless, early estimates of 20 to 30 parameters have been overtaken by more recent developments in linguistic theory. Roberts (2001) notes that Cinque's (1999) cross-linguistic analysis of adverb order implies that more than 64 parameters govern the extended projection of the verb within its clause. Guardiano & Longobardi (2003) estimate that there are "no less than 40 binary parameters (and, perhaps, no more than 50…)" for the internal structure of nominal phrases. Without putting a specific number on the totality of parameters for all aspects of syntax, it seems that (unless future research succeeds in shrinking the parametric inventory dramatically from its present level) the number of natural languages is more likely to be in the neighborhood of 2^{200} than 2^{20}. And 2^{200} is the sort of number that makes a difference to what is psychologically feasible. The arithmetic is straightforward. 20 independent binary parameters would yield 2^{20} grammars, which is over a million; 30 parameters yields over a billion; and every additional 10 parameters multiplies the number of grammars by a factor of more than 1,000. So 200 parameters would give a *very* large number of natural languages (vastly more than there are neurons in the human brain).[6]

Clearly, then, learning models *must* scale up; they must be efficient for large domains as well as for small ones. Specifically: the complexity of their operations should not increase in proportion to the number of grammars. Ideally it should stay in a range linked to the number of *parameters*: a function of 200, not of 2^{200}. This excludes any learning strategy which involves testing every grammar against each input sentence. Such methods are clearly out of the question. This is the reason why learning models that rely on grammar guessing have had to seek out clever search methods which can sample and test the vast field of grammars with great efficiency. Clark (1992) turned to genetic algorithms, which test batches of grammars, find the best ones in each batch, and then 'breed' them to obtain even better ones. Gibson & Wexler's TLA is a hill-climbing algorithm which proceeds through the domain from one grammar to another in search of improved performance at each step. These methods have not been robustly successful. For example, Kohl (1999) has documented a high failure rate for the TLA; see section 3.3 below. Clark (personal communication) does not regard a genetic algorithm, however successful, as a model of human learners because it exceeds plausible computational resource limits.

[6] If parameter values are conceived of as comparable to lexical items, acquiring 200 parameter values might be regarded as no more demanding than acquiring 200 new words, which is something young children do every month or so (and with the extra advantage that parameters are fully prefigured in UG). This is a tempting picture and worth considering seriously, but it must be borne in mind that new words often instantiate patterns already encountered, while a new parameter value can have a complex effect on the structures of the language.

Our own approach is quite different. The STL models, as discussed more fully below, can do some parametric decoding. Because human resource limitations preclude full parallel processing, this can be only partial decoding in the case of ambiguous sentences. Ideally a learner would identify *all* the grammars that could license a given sentence, but an STL can identify just *one* such grammar (and can sometimes tell that there are no others). The work is done by the sentence parsing mechanism, which is assumed to be innate. It finds one successful grammar for each sentence by drawing on parameter values on-line, as needed to complete the parse tree. This on-line processing is facilitated by casting the parameter values as treelets. Since these are transparently related to sentential parse trees, the parser can recognize which treelet(s) are needed to complete a parse. There is no guarantee that the one grammar that the parser identifies for an ambiguous sentence is the correct one, so this does not eliminate the ambiguity problem. But it does mean that the learner can focus its search on just those grammars that are real candidates (capable of licensing the current input), rather than sifting through the very much larger set of all grammars in the domain. This means that the task is scaled only by the extent of ambiguity in the domain, which is inescapable, and which will typically be less than fully exponential in the number of parameters.

2.5 Non-UG-based learners

Our simulation experiments to date have been almost exclusively concerned with parametric models that have access to a rich set of UG principles and a UG-determined list of parameters and their possible values. Even so, it is evident from the discussion above and the data reported below that syntax acquisition is not easy. We might imagine that without the benefit of UG it would be even more difficult. Yet there is a growing interest these days in learning systems that have little or no innate structure. Instead, they have powerful data processing techniques capable of picking up statistical regularities over a complex array of input. Dozens of papers advocating this approach have been presented or published in the last few years; see for example Lewis & Elman (2002), Pereira (2000), Seidenberg & MacDonald (1999) and references there. For those who believe that linguistic research has established that many properties of human language are universal, and hence most likely innate, this should be a matter of concern. From its inception, UG has been regarded as that which makes acquisition possible. But for lack of a thriving UG-based account of acquisition, UG has come to be regarded instead as an irrelevance or even an impediment. It is clearly open to the taunt: All that innate knowledge, only a few facts to learn, yet you can’t say how!

We believe the reason for this is the inability of recent UG-based acquisition models to *deliver* the rich information that UG contains, to the learner’s analysis of input sentences. To play its part, UG should interact productively with a novel sentence, guiding the learner to construct a legitimate representation of it, requiring grammatically relevant features to be included and irrelevant ones to be stripped off, assigning priorities to alternative analyses (a markedness ranking), and so forth. Otherwise, however richly structured it is, UG contributes little to acquisition. It defines an orderly set of grammars as the candidates, but that's all. It doesn't help the learner to fit the candidates to the data. As we have seen, an enormous amount of information that is contained in the combination of UG and a target sentence is simply wasted by learning procedures that can't unlock it. Suppose instead that we could develop a learning model that puts UG to work in the extraction of this information. Then the value of UG, and the folly of trying to tackle the task without it, might be more evident. In this regard, we believe the STL decoding models can help to provide a significant line of defense against encroaching empiricism in the theory of language acquisition. Nativist theories of human language will remain vulnerable until *some* UG-based learner is shown to perform well.

2.6 Summary

Parametric interaction and ambiguity make it difficult for a learner to extract from an input sample the information that it contains. Interaction and ambiguity occur in natural language on what is probably a grand scale, which renders impractical some otherwise imaginable procedures for setting parameters. Automatic triggering of parametric 'switches' is impossible, and it is uncertain at present what other mechanism could take its place. Note that these problems are not *created* by the learning mechanism. They are facts of the language domain, and no learning system, however well-designed, can make them go away. But some learning systems may cope with them better than others. Several recently proposed models have responded to the difficulties by giving up any attempt to read off the parametric signatures of sentences from their surface forms. They treat input sentences not as pointers to correct parameter values, but merely as the arbiters of grammar hypotheses selected in advance. The STL response to the failure of classical triggering is to salvage as much of it as possible, particularly the ability of the learner to *find* a licensing grammar for an input. Though resource limits preclude full decoding in case of ambiguity, even partial decoding radically reduces the scale of the task. It remains to be seen whether it narrows the search sufficiently to match the performance of human learners. This is what we want to know, and what we can begin to explore by means of the simulation experiments described below.

3. The simulation experiments

So far we have programmed 12 learning algorithms that have been proposed in the literature or are interesting variants of those. We have run each of them on the domain of more than 3,000 simplified but human-like languages. Every language serves as a target for every learning algorithm. A random sample of sentences of the target language is fed to the algorithm, which hypothesizes a grammar after each sentence. The trial stops when the target grammar is hypothesized, or after 100,000 inputs have been presented ("time-out"). For each learning algorithm we run 1,000 trials on each language; this is like setting 1,000 artificial 'children' to work on each language. We record the success rate, i.e., the percentage of trials on which the target was identified. We measure the average learning time, defined as the average number of input sentences a learner consumes before identifying the target grammar. An important combined measure of reliability and speed is the number of input sentences needed in order for 99% of trials (≡ 99% of 'children') to attain the target. Superset errors are excluded by fiat. We take it on faith that the Subset Principle will prevent these for all learning models (though implementing the Subset Principle faces surprising problems; see Fodor & Sakas, 2004). Learning counts as successful if the learner finds any grammar that licenses the input sample, even if it is not identical to the grammar that was used to generate the sample; in other words, weak equivalence with the target is the standard for convergence.

3.1 Design of the language domain

The language domain was designed to be large enough to reveal which models scale up well. Though tempted to add more parameters, for the present we have resisted expanding the domain further, in order to focus our resources on running the large-scale simulation tests of learning models described above. As the research project narrows in on the more successful models, its linguistic scope can be expanded. The input to learners is sentences as word strings, but all sentences in the domain have fully specified tree structures since this is essential to the functioning of the parser, which is especially important for structure-sensitive learners like the STL. The phrase structure format in which the grammars are expressed is useful because it allows rapid conversion into the operations of an effective parsing device. This is an integral aspect of the learning process; as parameter settings change, the learner must be ready to apply the new grammar right away to incoming sentences. However, since parameterized grammars were promoted within Government-Binding theory (more recently the Minimalist Program), our grammars assign structures to sentences which reflect fairly standard (though simplified) GB analyses.

There were painful decisions to be made about which linguistic phenomena to include and which to omit. To decide, we consulted adult speech to children in the CHILDES database (MacWhinney 2000).[7] We gave priority to syntactic phenomena which occur in a high proportion of known natural languages, which occur often in speech directed to 2-3 year olds, pose learning problems of theoretical interest, have a syntactic analysis that is broadly agreed on, and/or have been a focus of linguistic or psycholinguistic research. Following these

[7] Our research group has examined transcripts of adult speech to children learning English, French, German, Italian and Japanese. The children's age was approximately 1;6 to 2;6 years. Their MLU was very approximately 2; and the adults' MLU in child-directed speech was from 2.5 to 5.

criteria we included questions and imperatives, negation and adverbs, null subjects and null topics, verb movement to I and to C, preposition-stranding and pied piping, affix-hopping (though this hardly qualifies as widespread!), and Wh-movement. Table 1 lists the parameters. Their names indicate roughly what they do, though we lack space here to give details of how each one works in concert with the UG principles.

Table 1. The parameters that define the domain

Parameter	**Default**
Subject Initial [SI]	yes
Object Final [OF]	yes
Complementizer Initial [CI]	yes
V to I Movement [VtoI]	no
I to C Movement [ItoC]	no
Question Inversion (= I to C in questions only) [Qinv]	no
Affix Hopping [AH]	no
Obligatory Topic (vs. optional) [ObT]	yes
Topic Marking [TM]	no
Wh-Movement obligatory (vs. none) [Wh-M]	none
Pied Piping [vs. preposition stranding] [PI]	piping
Null Subject [NS]	no
Null Topic [NT]	no

Constraints on parameter value combinations (yielding 3,072 grammars, not 2^{13})

If [+ ObT]	then [- NS]	(A topic-oriented language does not have null subjects.)
If [- ObT]	then [- NT]	(A subject-oriented language does not have null topics.)
If [+ VtoI]	then [- AH]	(If verbs raise to I, no affix hopping.)

We need to be clear about what ended up on the cutting-room floor. We have as yet no scrambling, since its linguistic analysis is in a somewhat unsettled state at present. We have no DP-internal structure, though our CHILDES explorations have prepared the ground for that, and no overt Case marking or agreement. There is no clause embedding; all sentences are degree-0 like the great majority of sentences directed to children at this age. These limitations may seem stark, but they stand as an illustration of how little can be covered by 13 parameters.

As yet the domain includes no ellipsis and no discourse contexts to license sentence fragments, though such phenomena are extremely common in the input to children and may be among the earliest properties they are sensitive to. Our sentences are syntactic and pseudo-phonological entities only, with no semantics or LF representations. As a final disclaimer, we can almost guarantee that any syntactician's favorite analysis of the month will be absent. Our grammar does not employ feature checking in implementing the movement parameters

(Chomsky 1995 and since) and does not obey the Linear Correspondence Axiom of Kayne (1994 and since). It is possible that the outcomes of the simulation experiments would be different if the candidate grammars were more sophisticated in these ways. But to balance that is the fact that inaudible structure, undetectable to learners, cannot contribute to parameter setting; while if inaudible structure (such as non-lexicalized functional projections) is innate, it also will not engage the learning mechanism.[8] Finally, we must note some positive perks that our artificial learners enjoy, however unrealistically. As noted, lexical learning is not required of them since input sentences are realized by universal terminal symbols (S, Aux, O1, P, etc.). In effect, then, the learner knows all word categories and grammatical roles in advance. In real life such knowledge would be attained with some effort, perhaps through semantic bootstrapping and/or distributional learning (Pinker 1984). On the other hand, the input strings our learners receive contain at present no helpful cues to syntactic phrase boundaries such as might result from prosodic bootstrapping (Morgan et al. 1987).

3.2 Learning models tested

The purpose of all this is to find out whether any current learning models really work, in a domain with a realistic amount of parametric interaction and parametric ambiguity. Do they work reliably? Do they work as efficiently as child learners, and without exceeding the sorts of memory and processing resources available to children? We are also interested to know whether parameter decoding models work better than domain-search (grammar-testing) models; and within decoding models, whether guessing on an ambiguous sentence is a better or worse strategy than discarding it and waiting for unambiguous input.

In all models discussed here, learning is incremental in the sense that the learner hypothesizes a grammar (not necessarily different from its previous one) after each encounter with an input sentence. There is no memory for past inputs, so the data cannot be stored and mulled over later in search of generalizations. Except where noted, the learner is error-driven, i.e., if the currently hypothesized grammar $G_{current}$ can parse the sentence, it is retained. Changes are made only when $G_{current}$ fails. The models differ with respect to what the learner does next, when it has discovered that $G_{current}$ is wrong. The models we have tested can be grouped into systems that decode, such as the STL family, and grammar-testing

[8] Structure that is not overtly realized can cause learning problems even if it is innate, if other elements are parameterized with respect to it. For instance, the many adverb-related projections of Cinque (1999) are proposed as innate, but parameters determine where the subject and verb end up in that structure, creating potentially damaging ambiguity for learners concerning how to relate an overt word string to the inaudible structure (see Fodor 2001).

learners such as the TLA of Gibson & Wexler (1994) and the Variational Learner of Yang (2002).

The STLs include the Waiting-STL and a variety of Guessing-STLs. The Waiting-STL, known among ourselves as the 'squeaky clean' model, makes a grammar change only when it knows it is correct. It can therefore have confidence in the parameters it has set, and can use them to help in decoding the input for setting subsequent parameters.[9] In order to be able to do this it must be able to recognize and discard input that is parametrically ambiguous. This is possible, despite our assumption that the human parsing mechanism does not have the resources to parallel process all analyses of an ambiguous sentence. The human sentence parsing routines (on most standard assumptions) can tell when a choice point arises in the parse, signifying a local ambiguity in the analysis. To be on the safe side, the Waiting-STL treats every such local ambiguity as if it were a full ambiguity, and it refrains from setting parameters on the evidence of any part of the sentence that follows the choice point (Fodor 1998a). Thus it tends to overreact to ambiguity, but it never engages in guesswork. This learner needs unambiguous triggers to learn from, but these may be in short supply in natural language. The question of interest, therefore, is whether there is enough unambiguous input for all languages to be learnable. By contrast, the Guessing-STLs can learn from parametrically ambiguous input, because when they find a choice-point in the parse they simply guess between the analyses (Fodor 1998b). The Guessing-STLs differ from each other with respect to the specific principles that guide their guesses. Consider the parser at the point at which it has discovered that the current grammar cannot provide a full analysis of the sentence. It needs to pull in a new parameter value treelet to complete the parse tree. Suppose that more than one of the treelets that UG makes available would do the job. The *Any Parse* strategy tells the parser to choose between them at random. The *Minimal Connections* strategy picks the parameter value that gives the simplest tree (in accord with standard parsing principles such as Minimal Attachment and Late Closure). The *Least Null Terminals* strategy picks the parse with the fewest empty categories (equivalent to the Minimal Chain Principle for parsing). The *Nearest Grammar* strategy picks the grammar that differs least from $G_{current}$.

[9] The Waiting-STL's confidence in unambiguous triggers does not allow for the possibility of ungrammatical input, or grammatical input that is misconstrued. Since this does occur, however infrequently, any such deterministic (non-revising) learner is almost certainly too brittle to survive in the real world. However, learning models like the Waiting-STL that don't rely on guessing are of considerable theoretical interest, since children are often regarded (though perhaps only as an idealization) as setting parameters accurately without engaging in trial and error. Against this must be set the evidence from language change, and some psycholinguistic experiments, that acquisition errors occur.

Among grammar-testing learners, the TLA's properties are well-known. It responds to the failure of $G_{current}$ on an input sentence by changing the value of any one parameter, trying out the new grammar on the sentence and adopting it if the parse is successful. If the parse fails, the TLA retains its previous hypothesis. The restriction to one changed parameter value is the *Single Value Constraint*. Not adopting a new grammar if it can't parse the current input is enforced by the *Greediness Constraint*. A non-greedy version of the TLA was considered by Berwick & Niyogi (1996). This dispenses with the parse test, and simply adopts the new grammar. Berwick & Niyogi also contemplated a TLA without the Single Value Constraint. When $G_{current}$ fails, this tries out *any* other grammar and adopts it if it passes the parse test. There are also grammar-testing models which, unlike the TLA, have memory for the prior success or failure of each of the parameter values. In Yang's Variational Learner, a parameter value is strengthened if it participates in a grammar that successfully parsed an input sentence, and is weakened if it was in a grammar that failed to parse an input. Note that this reinforcement regime is only approximate because of the kind of interaction that Clark's work has drawn attention to. A good parameter value in an otherwise incorrect grammar is punished, and a wrong parameter value is rewarded if it gets a free ride in a grammar that is otherwise correct and doesn't need that value in order to license the current sentence. The Variational Learner is not error-driven. In a sense it has no $G_{current}$, but instead a graded success rating for each parameter. In selecting a grammar to try out on the next sentence it chooses parameter values with probability proportional to their current success weights. We have considered also an error-driven variant of this model (Sakas & Nishimoto, 2002) which is like Yang's original but has a $G_{current}$ consisting of the currently more successful value of each parameter. Only if that fails does the error-driven Variational Learner shift to a different grammar, selected on the basis of probabilities as above. Results for these Variational Learners are not included in the present paper; see Sakas & Nishimoto (2002) for relevant data.

Our experiments also include two models that serve as benchmarks against which to compare the others. These are not proposed as psychologically realistic models. One is too powerful to model human learning, and the other is too weak. The powerful one is the Strong-STL. It parallel-parses an input sentence, finds every grammar that could license it, and adopts all and only the parameter values that those grammars share, which are bound to be correct. The model that is too weak is an error-driven system similar to the TLA except that when $G_{current}$ fails it adopts any other grammar at random. Thus it samples the domain without any systematic search strategy. Though not worth considering as a psychological model, it is of interest because Berwick & Niyogi (1996) found this error-driven random learner to be superior to the TLA under certain circumstances.

3.3. Results

We note first some previous simulation results in the literature. Most relate to the TLA. Berwick & Niyogi (1996) found that in the Gibson & Wexler 3-

parameter domain, the TLA converged on the target more slowly than an error-driven random guessing learner. The TLA also fails to converge on the target in many cases, due to the problem of local maxima, discovered by Gibson & Wexler and discussed further by Berwick & Niyogi. Sakas (2000) reported, on the other hand, that the TLA performs better than the random model on strongly smooth domains (i.e., domains in which similar grammars yield similar surface languages, unlike what is thought to be typical of natural language). In an investigation of how well the TLA scales up to a more realistic domain size, Kohl (1999) reported a TLA failure rate of 95.4% on her domain of 2,304 languages. Kohl found also that no default (starting) grammar could avoid TLA learning failures on her domain; the best starting grammar succeeded only 43% of the time. She also observed that some TLA-unlearnable languages are quite natural, e.g., Swedish-type settings. As she noted, it would count in favor of a learning model if it failed on languages that, while apparently consistent with UG, are not attested. Concerning the Waiting-STL, Bertolo et al. (1997) noted that it is paralyzed by weakly equivalent grammars, i.e., distinct grammars which license the same set of surface sentences (word strings). Those sentences are parametrically ambiguous and so must all be discarded by the Waiting-STL, leaving it with no data at all for setting parameters. These findings are the background for our experiments, some of whose outcomes are more cheerful.

Table 2 shows failure rates and speed of learning for 10 different learning algorithms. Consider the failure rates first. Given the reliability with which children acquire the language(s) they are exposed to, any failure rate above zero is a disqualification. It might be objected that a learning model's failures are not inappropriate, but correspond to children with unexplained language deficits not associated with detectable neurological damage or other known factors. But failure rates of 70% or 80% clearly cannot be excused in such fashion. The TLA is ruled out on this ground, confirming Kohl's findings, and so are three STL variants. The failure of the Waiting-STL answers one of our questions above. It suggests that this domain does *not* contain a sufficient number of unambiguous triggers to set all parameters for all languages without resort to trial and error. This might change as future research adds more distinctive sentence types to the domain (see section 2.2). However, the fact that the relatively simple sentences of this domain cannot be learned by the waiting strategy suggests that even if learning were ultimately successful, it would get off to a very slow start. One source of hope is still open for this 'squeaky clean' non-guessing learner. We have not yet implemented the important distinction between ambiguous parameters and irrelevant parameters. The former participate in the derivation of a sentence and are successful whichever value they take; the latter do not participate in the derivation of the sentence at all. For example, a sentence consisting of just a subject and a verb is ambiguously derived, as Gibson & Wexler observed, by a subject-initial parameter setting, or by a subject-final setting plus the positive setting of the verb-second parameter. By contrast, that sentence does not need either setting of the parameter that orders the verb and object within VP; this parameter is simply irrelevant to the sentence. So is the parameter that controls preposition stranding versus pied piping, for a sentence with no XP movement or no prepositional phrase. In our results to date, parametric irrelevance is lumped together with parametric ambiguity; both cause a sentence to be discarded by the Waiting-STL without setting any parameters. This is clearly too drastic, and gives an unduly low estimate of the efficacy of an ambiguity-avoidance strategy. It remains to be seen how far the failure rate will fall when these cases are no longer counted against the waiting model. However, the improvement would have to be substantial if it is to rescue this approach. On current evidence, an obsession with perfect accuracy seems unlikely to be the best technique for learning a natural language. See Fodor (1998b) for other reasons for doubting the wisdom of such a strategy.

Table 2. Performance of 10 learning algorithms

Algorithm	**% failure rate**	**# inputs needed (99% of trials)**	**# inputs needed (average)**
Error-driven random	0	16,663	3,589
TLA original	88	16,990	961
TLA without Greediness	0	19,181	4,110
TLA without SVC	0	67,896	1,273
Strong-STL	74	170	26
Waiting-STL	75	176	28
Guessing-STLs			
Any Parse	0	1,486	166
Minimal Connections	0	1,923	197
Least Null Terminals	0	1,412	160
Nearest Grammar	80	180	30

The failure rate of the Strong-STL may be somewhat exaggerated, as in the case of the Waiting-STL. But taken at face value it reinforces the assessment that natural languages in at least some cases provide too little information to enable an incremental learner to establish all parameter values without resort to guessing. The Strong-STL is of interest because it does as well as any error-avoidance learner could. It can learn not only from unambiguous triggers but also from the unambiguous aspects of ambiguous triggers. For instance, if one sentence had a hundred distinct analyses but they all included Wh-movement, the Strong-STL could definitively adopt the positive value of the Wh-movement parameter. Yet even this power, it seems, does not suffice for reliable learning. The table shows that both the Strong-STL (which is not psychologically feasible) and the Waiting-STL (which is) are extremely fast learners when they are not hung up by ambiguity, yet ambiguity takes a serious toll: both models fail more often than they succeed. But these summary data hide an interesting difference between them. If we look at the time course of parameter setting within learning trials, we find that the Strong-STL (which does full parallel parsing) is considerably faster than the Waiting-STL (which does only serial parsing). For instance: For the language with French-type parameter settings (see section 3.4), after receiving 10 sentences the Strong-STL had set on average 9 parameters, while the Waiting-STL had set none. Clearly, the Strong-STL is much more efficient at extracting information from input. But it didn't *finish* the task any faster. Both algorithms took an average of 61 sentences to converge on the target. This is because after its rapid start, the Strong-STL hit a point where it couldn't advance for a long time. *This* hang-up must have been caused by a paucity of information in the input, not by an inability to extract it. In other words, this difficulty is inherent in the language domain, not the learning procedure, and this is an indication that human resources are not the limiting condition on language acquisition. Rather (contrary to all functional explanations), it may be that natural language design is not cooperative with *any* incremental learning algorithm, however powerful.

The Nearest Grammar version of the Guessing-STL also fails often, but for a different reason. The cause in this case, we believe, is not primarily ambiguity but something more like what causes local maxima in the TLA. The Nearest Grammar strategy is a conservative force, not unlike the SVC. The grammar is altered as little as possible at each step, in order to retain the fruits of past learning while accommodating new input. But excessive conservatism can keep the learner cycling through the same small set of grammars; even if the target is nearby, it is unreachable. The simulation data reinforces here a finding familiar in the machine learning literature: that exploration is sometimes a greater virtue than conservatism.

Consider learning rates now, for the models that succeed reliably enough to be worth considering. The error-driven random guessing system and the TLA without Greediness are quite similar, which shows how little work the SVC does by itself. By mathematical necessity, the random guess model requires approximately as many inputs on average as there are languages in the domain. The TLA with SVC but without Greediness does not do better than that. The TLA with Greediness but without the SVC does distinctly worse than that. Some Guessing-STLs (though not all) learn approximately 10 times faster than the TLA-related models. Thus, the ability to decode the parametric signatures of sentences lifts the learner into a higher level of performance. The three successful STL guessing strategies have quite similar outcomes, though two represent familiar parsing strategies while the other is random. The similarity may be due to the fact that the sentence structures in our domain are simple and fairly uniform, so the different selection strategies may not get a hold on them.

To summarize: The data show that not all models scale up well. The error-driven random guess model provides a baseline. It is slow, its efficiency being a simple function of the number of languages in the domain, which we flagged in section 2.4 as a danger sign. The TLA variants that reliably converge are no faster than the random guess baseline. The original TLA is also slow and fails to converge in many cases, as reported by Kohl. The ‘squeaky-clean’ decoding models (Strong-STL and Waiting-STL) fail often, presumably for lack of unambiguous triggers. Decoding models which guess in case of ambiguity emerge as the most efficient. They are tolerably fast and are free of errors. For these Guessing-STLs, on-line parsing strategies appear to make good learning strategies. Since they’re not notably better than random guessing this will need to be confirmed on more elaborate domains in future, but it is encouraging for the STL approach, which relies on the parser to sift through potential parameter values on-line to find ones that work. The data indicate that conservatism can increase learning speed but causes many errors even when grafted into a decoding model.

A fair conclusion overall is that learning-by-parsing fulfills its promise. (For other work that relates learning and parsing see Berwick, 1985, Seidenberg & MacDonald, 1999, and references there.) The STL approach sees learning as a natural consequence of a child’s desire to comprehend sentences s/he hears. The child’s parsing routines, like an adult’s, seek out aspects of the grammar that allow incoming words to be combined into a legitimate syntactic tree. The only difference for children is that the parser must be prepared on occasion to reach out beyond the current grammar and make use of new parameter values (treelets). This is an extremely natural psychological mechanism, and the simulation results show that it outperforms other models on a combination of reliability and learning speed. It is gratifying to discover that as computational models gain in psychological verisimilitude they become more, rather then less, efficient. It suggests we may be on the right track at last.

3.4. Further investigations: Comparing languages

Now that we have a workable learning strategy, we can make use of it to investigate other questions of interest. Holding the learner constant, we can ask whether some languages are easier than others; whether default parameter values help or hinder acquisition; whether overt morphological markings facilitate the setting of syntactic parameters; and so forth.

Table 3. Cross-language comparisons, for the Minimal Connections STL

Guessing-STL$_{\text{MinConn}}$	**# inputs needed (99% of trials)**	**# inputs needed (average)**	**# default parameter values**
‘Japanese’	87	21	8
‘French’	99	22	10
‘German’	727	147	7
‘English’	1,549	357	8

‘Japanese’ and ‘French’ are acquired more rapidly than the others, and indeed more rapidly than most other languages in the domain, since the number of inputs consumed in acquiring them is considerably lower than the average across all languages in Table 2. ‘English’ is the slowest of the four, but it is squarely average for the domain. We don’t know whether there is empirical support for differences in ease of acquisition among real languages. Observed differences might be attributable to cultural differences rather than grammatical ones. But it can be asked what makes a language easier in this bare learning situation where other factors do not intrude. The data are not predicted by how many of the target parameter settings are defaults. We suspect, though we have not yet confirmed this, that what matters most is parametric ambiguity, either in terms of the degree of overlap with neighboring languages, or as a lack of informative triggers. As noted above, it would be of interest to know whether the more difficult language types for learners are those not used by human societies. That is not inconsistent with our data, since the subject-final languages were learned more slowly on average than the humanly more frequent subject-initial languages (1127 versus 716 sentences for 99% convergence). But there are many reasons to be considered for why that might be so.

4. Sensitivity to input properties

Our simulation experiments test the ability of the learning model to extract grammar-relevant information from the perceptible properties of the input language sample. Learning efficiency should therefore be sensitive to the number and kind of distinctions that are overtly marked in the surface forms of sentences, though it is also dependent on the amount and kind of information that UG supplies; indeed, the interaction between these is what we see as the heart of UG-based learning. There is considerable theoretical interest in finding out to what extent UG-based parameter setting is input-paced (e.g., Evers & van Kampen 2001). From our experiments we can begin to create a profile of what input-paced learning looks like. If this does not match the acquisition sequence of child learners, that could suggest biological timing, such as late maturation of some parameters, or input filters of some kind.

With Carrie Crowther, we have begun a series of experiments in which, without altering the underlying grammars in any way, we manipulate the informativeness of the input sample by varying which syntactically relevant properties have overt ‘phonological’ realization. We can add to the original sentences some morphological markings of syntactic features such as Case, agreement, and finiteness to see how these affect the rate of learning. In real-life language learning, some languages provide these markers but some do not. Even for languages that do, a child must acquire them before getting any benefit from them. It seems plausible that they speed up syntax acquisition, but perhaps instead the morphological learning involved just creates more work which slows everything down. Another interesting addition to input sentences is the phonological realization of syntactic phrase boundaries provided by prosodic phrasing. Christophe et al. (2003) have shown that the prosodic patterns associated with a head-initial and a head-final language can be discriminated by infants at a very early pre-syntactic stage. Morgan et al. (1987) demonstrated the benefit of prosodic phrasing in the learning of simple constructed languages by adults. We can follow up these findings with simulation experiments on our language domain. The relation between syntactic and prosodic boundaries is imperfect because prosodic phrasing is affected by other factors such as phrase length and speaking rate. But even partial prosodic cues could help to disambiguate between competing syntactic tree structures for an input word string, which would constrain the grammar choices more tightly and improve learning. Prosody can also provide cues to illocutionary force, so learners can avoid confusing the syntactic characteristics of questions, declaratives and imperatives. Illocutionary force is often recognizable also from the semantics or the pragmatics of the discourse. Our language domain, without semantics or discourse contexts, originally provided no markers of illocutionary force except a KA complementizer for otherwise unmarked questions. A convenient substitute, for the purposes of our experiments, is the use of 'audible' features such as [ILLOC DEC] or [ILLOC IMP].[10]

Our first experiment compared learning rates when the finiteness feature on verbs was audible versus inaudible. Finiteness is predictable in our domain: UG requires the highest verb in every sentence to be finite and all others to be non-finite. (Recall that there are no embedded clauses.) However, the highest verb in an imperative sentence is obligatorily a phonologically null auxiliary, which means that the highest *audible* verb in an imperative is non-finite. Consequently, overt [+/-FIN] markings can distinguish declaratives from imperatives. This can be valuable, especially when no other cues to illocutary force are available (see above). In particular, a [-FIN] marking could head off an error in setting the null subject [NS] parameter. Imperatives have a null subject universally, even in [-NS] languages; the parameter is relevant only to declaratives and questions. So if a learner of a [-NS] language were to misinterpret an imperative sentence as a declarative, it would mis-set the parameter to [+NS]. Since null subjects are mutually exclusive with null topics in this domain, and null topics imply obligatory topicalization, this mistake could initiate a cascade of other errors. It is a reasonable prediction, therefore, that although finiteness is not itself parameterized, overt marking of finiteness could speed the correct setting of other parameters.

The experimental result showed no such improvement. This seems baffling until it is noted that the Subset Principle already precludes the potential [+NS] error. The Subset Principle requires a learner to treat a sentence that is

[10] The form of these notations is not important; other representations would do equally well. For instance, our universal grammar has a complementizer *PLEASE* for imperatives, which is silent and doing no work at present. We could have made this audible, and extended the KA complementizer to all questions, leaving unmarked sentences as declaratives.

ambiguous between imperative and declarative as an imperative. It does so because in our domain (which lacks expletives and has no ±pronominal contrast) the [NS] parameter is a subset/superset parameter. Setting it to the superset [+NS] value is always to be avoided if there is an alternative way of licensing the input. Since the imperative analysis doesn't require any marked parameter settings, it is preferred. This works for [-NS] languages, and it creates no problem for a genuine [+NS] target language since the correct setting will be triggered by other sentences which couldn't be imperative (e.g., sentences with an overt auxiliary or KA marking or a [+WH] element). A moral we have drawn from this little experiment is that we weren't smart enough to anticipate the behavior of the language domain even though we constructed it brick by brick ourselves. The reasoning failure was ours, but perhaps it is a hint that any deduction about interactions within a complex domain deserves to be empirically checked. Checking it by simulation is the closest we can approach to checking it against the real natural language domain, whose properties we don't control and don't fully know.

Our next experiment assessed the benefits of providing *direct* information about illocutionary force, in the form of [ILLOC] feature specifications. Some morphological marking of illocution occurs in natural languages, such as the interrogative complementizers -*ka* and -*no* in Japanese. Movement and deletion operations also distinguish illocutions, as in English. Prosodic marking is also common. Semantically, a Wh-constituent indicates a question. Prior to learning these form differences, we suppose that children tell whether an input is an imperative, declarative or question on the basis of the semantic and conversational context. Our [ILLOC] features can be regarded as a shorthand for this semantic knowledge. The illocutionary type of a sentence is relevant to several parametric phenomena (in addition to null subjects, as above), such as the difference between languages in which the finite verb always raises from I to C, like German, and languages where it does so only in interrogatives, like English. In terms of our parameters, this presents itself as a choice between [+ItoC] and [+Qinv] when the learner encounters a question with its verb in C. The [+ItoC] setting subsumes the [+Qinv] setting in the sense that the movement operation is the same in both but it applies in a broader versus a narrower context. This does not create a superset situation, because the movement is not optional; a language with only [+Qinv] has declarative sentences with the finite verb in I (or lower) that do not occur in a [+ItoC] language. So an incorrect choice would not be an incorrigible error. On the other hand, it could misdirect the learner for a while, and the Subset Principle can't step in to put it right. So again, it seemed reasonable to predict that learning becomes speedier when overt information is supplied. In this case, too, we were wrong. The results showed that when [ILLOC] is audible, learning is slower. How could this be?

We realized that the learner without audible [ILLOC] had an easier task. Without [ILLOC] there is a set of weakly equivalent grammars compatible with the input sample, such that hypothesizing any one of them counts as convergence on the target. (This corresponds to speakers having mental grammars which differ in ways that don't affect their observable language: an I-language difference with no E-language difference.) But with [ILLOC] features in sentences these grammars are no longer equivalent, so the learner must identify just one correct grammar. The equivalences don't involve just the [Qinv] and [ItoC] parameters discussed above, but most particularly the obligatory topic parameter. When illocutionary force marking is absent, [+ObT] languages and [–ObT] languages are sometimes surface identical (when other parameters are set the same way in both). Suppose the input sample was generated by the [-ObT] value. The [ObT] parameter applies to declarative sentences; imperatives and yes/no questions universally have no topic. The subset value [+ObT] requires all declaratives to have a topic, while [-ObT] also licenses declaratives with no topic. However, without ILLOC information, even [+ObT] could license all the target sentences, if declaratives without topics can be analyzed away as being imperatives or yes/no questions (i.e., when this is not ruled out by sentence properties such as KA or by the Subset Principle). In such cases the learner can converge regardless of which value of the [ObT] parameter it adopts. Of course, this would be a very strange 'child' who gets the word strings right but mistakes declaratives for questions. So these data underscore the importance of upgrading language domains for simulation research to include representations of sentence meaning. Real children learn not just the forms of sentences but pairings of form and meaning. See Villavicencio (2000), where LF representations are associated with surface syntactic structures.

Consider now a situation in which illocutionary force is known. Learning becomes more difficult because the learner can't get away with mixing up declarative and non-declarative sentences. The [-ObT] target language now includes some topicless sentences that are undeniably declarative, so the learner *must* posit [-ObT]; the default value [+ObT] will not do. With [ILLOC], then, the target is less broad and the learner must be more precise. But as long as the input provides the necessary information, why should that be hard? The answer is that the [-ObT] setting is particularly difficult to recognize. It is disfavored by the Subset Principle, so it needs triggers; but many of its potential triggers are ambiguous. A trigger would be any declarative sentence without a topic. Consider a verb-initial declarative. Since topics are sentence-initial (in Spec,CP which universally precedes C^0), this would seem to be a topicless sentence. But of course that is not a safe conclusion if the language might, for all the learner now knows, be a null topic language in which a topic can be present but inaudible. The Subset Principle would give [-ObT] the benefit of the doubt here (see below). But it would vote against [-ObT] in the case of declaratives with just an overt subject or object preceding the verb. These too are unreliable triggers for [-ObT]. Though the subject or object might be in its underlying position, it might instead have undergone string-vacuous topicalization (where there is no I-toC raising). Furthermore, no sentence with an initial Wh-phrase can be a trigger for [-ObT], because a fronted Wh-phrase occupies the Spec,CP position and masks whether there would otherwise have been a topic in that position. Evidence that the language has null subjects would be excellent evidence for [-ObT], since our UG prohibits [+NS] for topic-oriented languages, which it equates with [+ObT] languages. But the triggers for [+NS] are themselves not always unambiguous. All in all, therefore, reliable triggers for [-ObT] are rare. The Guessing-STL$_{\text{MinConn}}$ did eventually muddle its way through to the correct parameter settings for the ILLOC-marked sentences, but it was slow. The extra input information gave greater semantic precision, but in terms of learning speed it did more harm than good.

A third experiment examined the usefulness of a cue that could distinguish between the presence of a null topic and the absence of any topic. As noted above, the difference between these can be important for setting other parameters. In this experiment we made verb subcategorization information available to the learner. This is something that children must normally learn. We suppose they do so by bootstrapping from verb meaning or tallying distributional contexts – processes which are not included in our simulations. We simply supplied this information to the learning algorithm in the form of [SUBCAT] features, so that we could investigate the syntactic consequences of subcategory knowledge.

It might seem that every sentence displays a subcategorization of its verb, but this is not so if arguments can be missing. For learners who don't yet know the relevant parameter values, many sentences are ambiguous between an analysis in which an argument was present but deleted (phonologically null), and an analysis in which no such argument was present at all. In our domain the only way to 'delete' a constituent is via the positive value of the null topic parameter, [+NT], which allows any element to be null if it is in topic position. A null topic, like any other topic, is associated with a trace elsewhere in the sentence. It does not appear overtly in either position. If it is an optional element, such as an adjunct, the word string gives the learner no indication of its presence in the tree.

Consider a verb-initial declarative sentence. This clearly has no *overt* topic. The learner is faced with a choice between two marked parameter settings: [-ObT] or [+NT]. That the marked value of the [ObT] parameter is [-ObT] may seem perverse linguistically, but it is mandated by the Subset Principle, which must always favor obligatory over optional phenomena. For the same reason, the marked value for null topic is [+NT]. Of interest here is that the Subset Principle selects between the two parameters. It requires the learner to adopt [-ObT] rather than [+NT], since the latter generates a more inclusive language. [+NT] licenses sentences with an indirect object but no direct object (e.g., *Subj Verb O2*), and in a preposition-stranding language it licenses sentences with a preposition but no O3 that could be its object (e.g., *Subj Verb P Adv*). Neither of these is normally permitted[11], and neither can occur as a result of topicalization being optional, i.e., [-ObT]. So these sentences would be the triggers for setting [+NT]. But they are

[11] This presupposes that prepositions are distinguishable from particles. In our domain there are no particles (or intransitive prepositions), but in natural languages this is another ambiguity that the learner would have to resolve.

not abundant. Especially in a pied-piping language, the learner might wait a long time before hearing evidence of [+NT]. Verb SUBCAT information could supplement this meager evidence. Only [+NT] could account for the surface absence of the direct object of a transitive verb, the indirect object of a ditransitive verb, the locative argument of a verb like *put*, and so on. Thus, knowledge of obligatory verb-argument structures could be valuable to the learner for making the fine distinction between null constituents and non-existent ones.[12] We predicted, therefore, that learning would be faster when verbs were supplied with SUBCAT features. And in this case the data did support the prediction. There were no unforeseen interactions with other parameters, no damaging side-effects. The subcategory information was doing some real work.

To summarize: Enriching the input has complex effects on learning. Richer input is *beneficial* if it provides information about something that must be learned anyway, and especially if other cues are scarce. Richer input can be a *hindrance* if it creates a distinction that otherwise could have been ignored. The specific outcomes of these experiments obviously depend on the properties of the particular domain, so they cannot be generalized as they stand. But the domain can be tailored as necessary to issues of interest, so other investigations are possible. The ultimate interest of studies like these is the light they can shed on child language acquisition. We can accumulate case studies of learning with and without the aid of various cues in the input, we can vary the frequency of these cues, and we can see how the course of learning differs as the weight of decision-making falls more on UG or more on evidence in the input. Eventually we hope to be able to compare these data with the course that children take and draw some conclusions for currently debated issues such as poverty or non-poverty of the stimulus, whether sensitivity to frequency statistics in the input implies lack of UG guidance, and so forth. We can't do this yet, but we do believe that proponents of UG-based learning must face up to these difficult matters as soon as possible.

5. Future directions

[12] This is part of a more interesting interplay between lexical and syntactic learning. For instance, SUBCAT information could help learners identify trace positions for Wh-movement. Once Wh-movement is acquired, a learner should be able to tell that a novel verb is transitive even if it has no object in the VP, if there is a fronted Wh-NP and no other possible position for its trace. What we have left out of the discussion above is the question of whether a verb previously encountered as transitive, but now occurring without an overt object, would be preferentially construed as having a null object or as being optionally intransitive. Is it the syntax or the lexicon that is expanded? Is it semantics that makes this decision, or a general learning principle?

These comparisons of learning models and input properties are just the beginning of what the language domain enables us to do.[13] Next on our agenda are experiments to assess how vulnerable different learning models are to 'noisy' input. We will re-run the simulations with 1 sentence in every 5, or every 10, or every 100, drawn at random from a language other than the target, either from a single language or from different ones. We will also investigate a version of the 'starting small' hypothesis (Newport, 1990; Elman, 1993), by seeing how much facilitation results from presenting mostly short sentences early on. And we will use our cross-linguistic survey of the sentence patterns in child-directed speech to find out how much facilitation (if any) derives from the exact mix of sentence types in the sample of language that children hear (cf. Newport, 1977; Yang, 2002).

Continuing the evaluation of learning models, we are now adding connectionist and statistical learners to see how they fare with respect to speed and reliability. Allowance must be made for the fact that they have a far more substantial task to do than learners that are already in possession of large parts of the grammar from the outset, so some careful thought must go into how the comparisons can be fairly drawn. We also plan to refine the STL models. In particular we are eager to add the STL variety that we believe comes closest to the psychological truth. This is the "Parse Naturally" STL of Fodor (1998b), which employs nothing but the standard adult parsing strategies and an innate lexicon of parameter value treelets whose weights are continuously adjusted (not unlike the error-driven version of Yang's Variational Learner sketched above).

We also plan to make use of the time-course data gathered from every learning trial (each of 1,000 'children' for each of 3,072 target languages), in two ways. First, it can be used for a more fine-grained evaluation of learning models with respect to whether they set the parameters in a realistic sequence which matches that of children. Second, a massive comparison of all of these time courses could reveal whether there is an *optimal* sequence for setting parameters. We can contrast sequences which result in very rapid convergence on the target with those which are the slowest. If there are some sequences associated with notably superior performance, this could reinforce learning models that assume an innately prescribed schedule (e.g., Dresher, 1999; Roeper & de Villiers, 1992).

This will keep us busy for a while to come. But other ideas are very welcome. The language domain is accessible at www.colag.cs.hunter.cuny.edu

[13] There are things this project is not equipped to do, the most notable being language production. What do learners *say* when they haven't fully mastered the grammar? We can examine the sentences generated by a learner's incorrect intermediate grammars. Some are correct sentences of the target language, some are not, and some target sentences are missing. But the non-target sentences don't at all resemble baby-talk, e.g. the artificial equivalent of *Me has Mary not kissed why?*, or later on: *Whom must not take candy from?* Clearly we are missing some factors that shape children's output.

and we will be happy to provide assistance in using it to explore other hypotheses of interest. As we have begun to discover, the outcomes are not always as anticipated.

References

Bertolo, S., Broihier, K., Gibson, E., and Wexler, K. (1997) Cue-based learners in parametric language systems: Application of general results to a recently proposed learning algorithm based on unambiguous 'superparsing'. *19th Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Berwick, R. C. (1985) *The Acquisition of Syntactic Knowledge*. MIT Press, Cambridge, MA.

Berwick, R. C. and Niyogi, P. (1996) Learning from Triggers. *Linguistic Inquiry*, 27(2), 605-622.

Briscoe, E. J. (2000) Grammatical acquisition: Inductive bias and coevolution of language and the language acquisition device, *Language* 76(2), 245-296.

Chomsky, N. (1981) *Lectures on Government and Binding*. Foris Publications, Dordrecht.

Chomsky, N. (1995) *The Minimalist Program*. Cambridge MA: MIT Press.

Christophe, A., Nespor, M., Guasti, M. T. and Van Ooyen, B. (2003) Prosodic structure and syntactic acquisition: The case of the head-direction parameter. *Developmental Science* 6:2, 211-220.

Cinque, G. (1999) *Types of A'-Dependencies*. MIT Press, Cambridge, MA.

Clark, R. (1988) On the relationship between the input data and parameter setting. *NELS 19*, 48-62.

Clark, R. (1992) The selection of syntactic knowledge, *Language Acquisition 2, 83-149.*

Dresher, E. (1999) Charting the learning path: Cues to parameter setting. *Linguistic Inquiry* 30.1, 27-67.

Elman, J. L. (1993) Learning and development in neural networks: The importance of starting small. *Cognition*, 48, 71-99.

Evers, A. and van Kampen, J. (2001) E-language, I-language and the order of parameter setting. Unpublished ms. Utrecht Institute of Linguistics OTS.

Fodor, J. D. (1998a) Unambiguous triggers, *Linguistic Inquiry* 29.1, 1-36.

Fodor, J. D. (1998b) Parsing to learn. *Journal of Psycholinguistic Research* 27.3, 339-374.

Fodor, J. D. (2001) Setting syntactic parameters. In M. Baltin and C. Collins (eds.) *The Handbook of Contemporary Syntactic Theory*, Blackwell Publishers, Oxford, UK.

Fodor, J. D. and Sakas, W. G. (2004) The Subset Principle in syntax: The cost of compliance. Unpublished ms., City University of New York.

Gibson, E. and Wexler, K. (1994) Triggers. *Linguistic Inquiry* 25, 407-454.

Guardiano, C. and Longobardi, G. (2003) Parametric syntax as a source of historical-comparative generalisations. Unpublished ms., Pisa-Trieste.

Jain, S., Osherson, D., Royer, J. and Sharma, A. (1999) *Systems that Learn*, Second Edition. MIT Press, Cambridge MA.

Kayne, R. S. (1994) *The Antisymmetry of Syntax*. Cambridge MA: MIT Press.

Kohl, K. T. (1999) *An Analysis of Finite Parameter Learning in Linguistic Spaces*. Master’s Thesis, MIT.

Lewis, J. D. and Elman, J. L. (2002) Learnability and the statistical structure of language: Poverty of stimulus arguments revisited. In B. Skarabela et al. (eds.) *Proceedings of BUCLD* 26, Cascadilla Press, Somerville, MA.

Lightfoot, D. (1991) *How to Set Parameters: Arguments from Language Change*. MIT Press, Cambridge, MA.

MacWhinney, B. (2000) *The CHILDES Project: Tools for Analyzing Talk*. 3rd Edition. Vol. 2: The Database. Lawrence Erlbaum Associates, Mahwah, NJ.

Morgan, J. L., Meier, R. P. and Newport, E. L. (1987) Structural packaging in the input to language learning: Contributions of prosodic and morphological marking of phrases to the acquisition of language. *Cognitive Psychology* 19, 498-550.

Newport, E. L. (1977) Motherese: The speech of mothers to young children. In N. J. Castellan et al. (eds.) *Cognitive Theory*, Vol. 2. Lawrence Erlbaum, Hillsdale, NJ.

Newport, E. L. (1990) Maturational constraints on language learning. *Cognitive Science, 14,* 11-28.

Nyberg, E. H. (1992) *A Non-Deterministic, Success-Driven Model of Parameter Setting in Language Acquisition,* PhD dissertation, Carnegie Mellon University.

Pereira, F. (2000) Formal Theory and Information theory: Together again? *Philosophical Transactions of the Royal Society*, Series A 358, 1239-1253.

Pinker, S. (1984) *Language Learnability and Language Development*, Harvard University Press, Cambridge MA.

Roberts, I. (2001) Language change and learnability. In S. Bertolo (ed.) *Language Acquisition and Learnability*. Cambridge, UK: Cambridge University Press.

Roeper, T. and de Villiers, J. (1992) Ordered decisions in the acquisition of Wh-questions. In J. Weissenborn, H. Goodluck and T. Roeper (eds.) *Theoretical Issues in Language Acquisition: Continuity and Change in Development*. Lawrence Erlbaum, Hillsdale, NJ.

Sakas, W. G. and Fodor, J. D. (2001) The Structural Triggers Learner. In S. Bertolo (ed.) *Language Acquisition and Learnability*. Cambridge University Press, Cambridge, UK.

Sakas, W. G. (2000) *Ambiguity and the Computational Feasibility of Syntax Acquisition,* PhD Dissertation, City University of New York.

Sakas, W. G. and Nishimoto, E. (2002). Search, structure or heuristics? A comparative study of memoryless algorithms for syntax acquisition. *24th Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Seidenberg, M. S. and MacDonald, M. C. (1999) A probabilistic constraints approach to language acquisition and processing. *Cognitive Science* 23(4), 569-588.

Villavicencio, A. (2000) The acquisition of word order by a computational learning system. *Proceedings of CoNLL-2000 and LLL-2000*, 209-218, Lisbon, Portugal.

Yang, C. D. (2002) *Knowledge and Learning in Natural Language.* Oxford University Press, Oxford, UK.