

Running head: HOW RECURSIVE IS LANGUAGE?


How recursive is language? A Bayesian exploration

Amy Perfors
School of Psychology
University of Adelaide
Level 4, Hughes Building
Adelaide, SA 5005 Australia
Email correspondence to: amy.perfors@adelaide.edu.au

Josh Tenenbaum and Edward Gibson
Department of Brain & Cognitive Science
Massachusetts Institute of Technology
Cambridge, MA 02139 USA

Terry Regier
Department of Psychology
University of Chicago USA

How recursive is language? A Bayesian exploration

Recursion involves an inherent tradeoff between simplicity and goodness-of-fit: a grammar with recursive rules might be simpler than one without, but will predict the sentences in any finite corpus less exactly. As a result, one cannot conclude that any particular grammar or grammatical rule is recursive, given a corpus, without some way to quantify and calculate this tradeoff in a principled way. We present a Bayesian framework for performing rational inference that enables us to quantitatively evaluate grammars with and without recursive rules and normatively determine which best describe the sentences in a corpus of child-directed spoken English. Our results suggest three main points. First, they suggest that rational principles would favor a grammar with a specific type of recursive rule, even if there are relatively few instances of particular recursively-generated sentences in the input. Second, they suggest that the optimal grammar may occupy a representational middle ground between fully recursive and non-recursive. Finally, our results suggest that the optimal grammar may represent subject NPs distinctly from object NPs.  suggest that our method and insights can be usefully applied to address other questions in linguistics and the study of recursion.

1. Introduction

One of the most notable features of human language is its capacity to generate a potentially infinite number of possible sentences. Because such a capacity must result from an underlying generative mechanism (a grammar) that is recursive in some way, many linguists have concluded that recursion must be a fundamental, possibly innate, part of the language faculty (Chomsky 1957). Some have gone further and claimed that the core mechanism underlying recursion is the only part of language that is specific to humans (Hauser, Chomsky, and Fitch 2002). While the latter, stronger claim is contested (Pinker and Jackendoff 2005), the former has been largely accepted for decades. However, recent work on Pirahã, a language spoken in the Amazon basin, suggests that there may be a language that does not in fact contain any recursion in its phrase structure whatsoever (Everett 2005).

The empirical claim about Pirahã is the subject of much debate (Nevins, Pesetsky, and Rodrigues 2007; Everett 2007), and an essential key to resolving the debate is to be able to objectively determine whether Pirahã is better described by a grammar with recursive elements or by one without. But what does it mean to say that one grammar constitutes a “better description” of a language than another? In this article we argue that this question cannot be answered for any particular language without a rigorous, quantitative, and principled mechanism for comparing grammars with respect to linguistic corpora. We propose such a method and demonstrate its efficacy by applying it to a corpus of child-directed speech in English. Our results suggest that simple rational principles support the inference of a specific set of recursive phrase structures in English, even if few sentences in the corpus contain multiple embeddings resulting from multiple expansions of those productions. The method yields several insights about the possible role, nature, and learnability of recursion in language, and may ultimately be applicable to addressing the case of Pirahã.

1.1. Measuring recursion

A grammar is recursive if it contains at least one rule whose expansion can eventually involve a call to itself. There are three main types of recursion: left-branching, right-branching, and center embedding. The first two types of recursion occur in both finite-state (FS) and context-free grammars (CFGs), but center embedding exists only in grammars capable of representing phrase structure: that is, grammars whose formal complexity is at least equivalent to that of a CFG. A phrase of depth n results from a recursive rule that is expanded n times. For instance, the following center-embedded sentences are generated by self-embedding NPs with relative clauses.

- (1) (a) The cat eats [depth 0]
- (b) The cat that the dog chased eats [depth 1]
- (c) The cat that the dog that the boy petted chased eats [depth 2]

Human learners have difficulty parsing sentences with multiple center embeddings (Miller and Chomsky 1963; Marks 1968; Bach, Brown, and Marslen-Wilson 1986), probably due to performance factors such as limitations on working memory (Chomsky 1956; Miller and Chomsky 1963; Gibson 1998; Weckerly and Elman 1992). It is in principle possible that this particular English grammatical rule is not truly recursive: as long as there are no sentences with more than n embeddings, a grammar with n non-recursive rules would also parse the language. Indeed, work by Christiansen and Chater (1999) suggests that it may not be necessary to assume unbounded recursion in the grammar in order to qualitatively explain aspects of human performance.

How might a linguist or a child attempting to acquire the proper grammar determine whether a grammar, or a particular grammatical rule, is recursive? The standard argument that grammars of natural language must contain recursion dates back to Chomsky (1957), who pointed out that natural language appears to have the property of discrete infinity: it is composed of discrete basic elements (words) which can be combined to produce apparently infinitely many sentences. An infinite set can only be generated from a finite grammar if the grammar contains some form of recursion. But is it true that natural language is infinite? After all, there are no infinitely long sentences, and only a finite number of sentences have ever been uttered. Since any finite language may be captured by a finite grammar without recursive rules, why believe that grammars of natural language have recursive rules? The standard reason is simplicity: a non-recursive grammar capable of generating natural language would be very large, since it would require additional sets of rules for each additional depth of recursive expansion. Any evaluation metric that favors shorter (i.e., simpler) grammars should therefore prefer a grammar with recursive rules over one with non-recursive rules.¹


This simplicity-based argument is reasonable, but not airtight: it is qualitative, not quantitative, based on our intuitions about how much more complex a grammar with non-recursive instead of recursive rules would be. The complexity of a grammar would increase with each additional rule, and how many non-recursive rules would be necessary depends on the precise sentences in the corpus. Furthermore, it is the tradeoff between the complexity of a grammar and how well that grammar explains the observed sentences that is important – not its complexity alone. If choosing the simplest grammar were the only consideration necessary, a learner would always favor the grammar that can generate any sentence whatsoever. Such a

grammar is maximally simple since it contains no rules constraining how the terminal symbols may be combined, but we would obviously like to rule it out.

This suggests that a rational learner should evaluate a grammar with respect to how precisely it predicts the sentences observed as well as its simplicity. All else being equal, a grammar that generates sentences not present in the observed language should be dispreferred relative to one that does not. Unfortunately, recursive productions hurt the fit of a grammar on any finite corpus, since they will always predict sentences that are not observed. The fewer sentences in the corpus that result from multiple expansions of recursive rules, the more a grammar with recursive rules, relative to one without, will overgeneralize, since it predicts very long sentences that are never actually used.

Thus, recursion involves an inherent tradeoff between simplicity and degree of fit: while a grammar with recursive rules might be shorter and simpler than one without, it will also fit any finite set of sentences less precisely. Furthermore, both the degree of complexity and the degree of fit of the grammar depend on the precise nature of those sentences. As a consequence, we cannot conclude on *a priori* grounds that any grammar for natural language must contain recursion. While that may be a reasonable starting assumption, it might not be true in all cases, whether for a specific language (e.g., Pirahã) or for a specific rule or set of rules (e.g., center-embedded relative clauses in English). In order to evaluate these specific cases it is necessary to quantify degrees of simplicity and fit, as well as to calculate the tradeoff between the two in a principled and rigorous way.

How can we perform such a calculation? Traditional approaches to formal language theory and learnability are unhelpful because they presume that a learner does not take either simplicity or degree of fit into account (Gold 1967). A Bayesian approach, by contrast, provides an intuitive and principled way to calculate the tradeoff between a grammar's simplicity (prior probability) and its degree of fit to a corpus (likelihood). Such an approach is consistent with Chomsky's formulation of the problem of language learning, which presumes both a hypothesis space of grammars and the existence of an evaluation metric based on simplicity (Chomsky 1965). Indeed, it has been formally proven that an ideal learner incorporating a simplicity metric will be able to predict the sentences of the language with an error that approaches zero as the size of the corpus goes to infinity (Solomonoff 1978; Chater and Vitanyi 2006); in many more traditional approaches, the correct grammar cannot be learned even when the number of sentences is infinite (Gold 1967). However, learning a grammar (in a probabilistic sense) is possible, given reasonable sampling assumptions, if the learner is sensitive to the statistics of language (Horning 1969).

In this article we propose a Bayesian framework for grammar induction, which can be used to quantitatively evaluate grammars with respect to one another by calculating an ideal tradeoff between simplicity and goodness-of-fit. Because the framework combines  statistical inference mechanisms that can operate over structured representations of knowledge such as generative grammars, it can in principle apply to many interesting linguistic questions regarding representation and learnability. For instance, other work involving this framework suggests that structure-dependence in language need not be innate: a rational learner could infer based on typical child-directed input that language is better captured by a hierarchical phrase-structure grammar than by a finite-state one (Perfors, Tenenbaum, and Regier, under review).

Here we apply this framework to the issue of recursion by evaluating grammars of English with and without a particular set of recursive rules (NPs with relative clauses) as they apply to a specific corpus of actual spoken speech. Is a language with recursive NPs preferred by a rational learner over one without, even when the input does not contain sentences with many levels of embedding? Our results yield some surprising insights into the circumstances under which recursive rules might occur, and suggest that grammars that occupy a “representational middle ground” between fully-recursive and only non-recursive rules may be most preferred.



2. Method

We cast the problem of grammar induction within a Bayesian framework in which the goal is to infer which grammar G is most likely to have generated some data D (a corpus of child-directed speech). The framework assumes a probabilistic generative model for linguistic utterances, which can then be inverted by the learner to infer aspects of the generating grammar from the language data observed. A linguistic corpus is assumed to be created by first picking a grammar G from a space of possible grammars, and then by generating D from the specific grammar G by drawing from the conditional distribution $p(D|G)$. The inferences we can make from the observed data D to the specific grammar G are captured by the posterior probability $p(G|D)$, computed via Bayes' rule:

$$p(G|D) \propto p(D|G)p(G). \quad (1)$$

This equation states that the posterior probability of a grammar given the data ($p(G|D)$) is proportional to its prior probability ($p(G)$) times the likelihood of the data given the grammar ($p(D|G)$). The prior for a grammar $p(G)$ is calculated assuming a generative model of grammars (a grammar for generating grammars) that assigns higher prior probability to simpler grammars. The likelihood $p(D|G)$ reflects the probability of the corpus D given grammar G ; it is a measure of how well the grammar fits the corpus data. The posterior probability $p(G|D)$ thus automatically seeks a grammar that optimally balances the tradeoff between complexity (prior probability) and fit to the data (likelihood). In the following subsections we describe the specific grammars G , the corpus of data D , and the particulars of the computational framework.²

2.1. The grammars

Each grammar consists of a set of production rules specifying how one non-terminal symbol (the left-hand side of the rule) in a string may be rewritten in terms of two other symbols, terminal or non-terminal. Each rule is associated with a probability, such that the probabilities of all rules with the same left-hand sides (i.e., all S rules, all NP rules, etc.) add to one and the probability of a complete parse is the product of the probabilities of the rules involved in the derivation. All grammars are context-free, since CFGs generate parse trees with hierarchical structure and are often adopted as a first approximation to the structure of natural language (Chomsky 1959). Probabilistic context-free grammars (PCFGs) are also standard tools in computational linguistics (Jurafsky and Martin 2000; Manning and Schütze 1999).

Although PCFGs do not capture the full complexity of natural language, we work with them because they are complex enough to allow for an exploration of the role of recursion, because they are easily defined in probabilistic terms, and because CFGs have been the focus of recent debate over the role of recursion as the allegedly uniquely human core of language (Hauser, Chomsky, and Fitch 2002; Fitch and Hauser 2004; Gentner et. al. 2006). All of our

grammars can parse all of the sentences in the corpus, and all contain standard structures based in linguistic theory (including noun, verb, and prepositional phrases).

We evaluate three main grammars that differ from each other only in whether some rules are recursive or not. The simplest grammar is the fully recursive R-CFG, which contains recursive noun phrases such as $[NP \rightarrow NP CP]$ and is made up of 100 rules and 14 distinct non-terminals. Since the reason recursive rules are costly is because they do not fit the data precisely, we design our other grammars so as to minimize overgeneralization. One grammar accomplishes this by eliminating recursive rules entirely; the other decreases the weight assigned to them by creating identical “shadow” rules that correspond to non-recursive expansions. For instance, a grammar with shadow rules retains a recursive rule like $[NP \rightarrow NP PP]$ but also contains $[NP \rightarrow NN PP]$, where NN only expands into simple noun phrases such as $[NN \rightarrow Det N]$. Rules in this grammar with right-hand-sides containing NP (such as $[VP \rightarrow V NP]$) are therefore also joined by rules containing NN (such as $[VP \rightarrow V NN]$). The grammar containing both recursive rules and non-recursive “shadow” rules is B-CFG (“B” for both: 126 productions, 15 non-terminals).

The grammar without any recursive rules at all is identical to B-CFG except that the recursive NP productions are eliminated and replaced with multiply-embedded non-recursive productions involving an additional new non-terminal, N2. This depth-limited non-recursive grammar, N-CFG, therefore only parse sentences with up to two nested relative clauses. Sample rules from [Leahy](#) grammar are shown in Table 1.³

These three grammars do not capture the distinction between subject and object noun phrases, which have different syntactic properties, including differences in the difficulty of processing multiple recursive expansions. Recursive expansions of subject noun phrases (NP_S) results in the hard-to-understand center-embedded sentences discussed earlier, but sentences with recursive expansions of object noun phrases (NP_O) are significantly easier: compare *The cat that the dog that the boy petted chased eats* to *The boy petted the dog that chased the cat that eats* (Miller and Chomsky 1963). We can examine the role of recursion in each type of NP by creating additional grammars like our three basic ones, except that they contain rules with distinct NP_S and NP_O left-hand sides. Nine (3^2) grammars result from all combinations of the three kinds of recursion (recursive-only (R), depth-limited (D), and both (B)) and two noun phrases (NP_S and NP_O).

2.2. The corpus

The corpus consists of the sentences spoken by adults in the Adam corpus (Brown 1973) of the CHILDES database (MacWhinney 2000). Because grammars are defined over syntactic categories, each word is replaced by its syntactic category.⁴ For reasons of tractability, ungrammatical sentences and the most grammatically complex sentence types are removed from the corpus.⁵ The final corpus contains 21671 individual sentence tokens corresponding to 2336 unique sentence types, out of 25755 tokens in the original corpus, and includes interrogatives, wh-questions, relative clauses, prepositional and adjective phrases, command forms, and auxiliary and non-auxiliary verbs.

2.3. The probabilistic model

Grammars were scored using a probabilistic scoring criterion based on Bayes' rule, which combines the prior probability of a grammar G with the likelihood that the corpus D was generated by that grammar.

2.3.1. Scoring the grammars: prior probability.

A simplicity metric on a grammar may be derived in a principled way by defining the process by which the grammars themselves can be generated and then calculating each grammar's relative prior probability based on the distribution imposed by that process (Horning 1969). This process is a grammar-grammar, a generative model for grammars, in which each grammar is generated by making a series of choices. Longer grammars, which require more choices, will have a lower prior probability.

If one were generating a grammar from scratch, one would first need to choose the number of non-terminals n , and for each non-terminal k to generate some number R_k of rules, each of which is associated with a production probability parameter θ_k . Each rule i also has N_i right-hand side items, and each of those items must be drawn from the grammar's vocabulary V (set of non-terminals and terminals). If we assume that each right-hand side item of each rule is chosen uniformly at random from V , the prior probability is given by:

$$p(G|T) = p(n) \prod_{k=1}^n p(R_k) p(\theta_k) \prod_{i=1}^{R_k} p(N_i) \prod_{j=1}^{N_i} \frac{1}{V}. \quad (2)$$

We model the probabilities of the number of non-terminals $p(n)$, rules $p(R_k)$, and items $p(N_i)$ as selections from a geometric distribution, which assigns higher probabilities to lower numbers; production-probability parameters $p(\theta_k)$ are sampled from a discrete approximation of a uniform distribution appropriate for probability parameters (Dirichlet). This prior gives higher probability to simpler grammars – those with fewer non-terminals, productions, and items. Because of the small numbers involved, all calculations are done in the log domain.

2.3.2. Scoring the grammars: likelihood.

Inspired by the work of Goldwater, Griffiths, and Johnson (2006), the likelihood is calculated assuming a language model that is divided into two components. The first component, the grammar, assigns a probability distribution over the potentially infinite set of syntactic forms that are accepted in the language. The second component generates a finite observed corpus from the infinite set of forms produced by the grammar, and can account for the characteristic power-law distributions found in language (Zipf 1932). In essence, this two-component model assumes separate generative processes for the allowable *types* of distinct sentences (defined here as sequences of syntactic categories) in a language and for the frequency of specific sentence *tokens*. The probabilistic grammar is only directly involved in generating the allowable types.

One consequence of this approach is that grammars are analyzed based on individual sentence types rather than on the frequencies of different sentence tokens. This parallels standard linguistic practice: grammar learning is based on how well each grammar accounts for the types of sentence forms rather than their frequency distribution. Since we are concerned with grammar

comparison rather than corpus generation, we focus in this work on the first component of the model. We thus take the data to consist of the set of sentence types (distinct sequences of syntactic categories) that appear in the corpus, and we evaluate the likelihoods of candidate probabilistic grammars on that dataset.

The likelihood assigned to a grammar can be interpreted as a measure of how well the grammar fits or predicts the data. The penalty for overly general or flexible grammars is computed in the parsing process, where we consider all possible ways of generating a sentence under a given grammar and assign probabilities to each derivation. The total probability that a grammar assigns over all possible sentences must sum to one, and so the more flexible the grammar, the lower probability it will tend to assign to any one sentence.

More formally, the likelihood $p(D|G)$ measures the probability that the corpus data D would be generated by the grammar G . If we assume that each sentence type is generated independently from the grammar, this is given by the product of the likelihoods of each sentence type S_i , as shown in Equation 3. If there are M unique sentence types in the corpus, the corpus likelihood is given by:

$$p(D|G) = \prod_{i=1}^M p(S_i|G). \quad (3)$$



The probability of any sentence type S_i given the grammar $p(S_i|G)$ is the product of the probabilities of the productions used to derive S_i . Thus, calculating likelihood involves solving a joint parsing and parameter estimation problem: identifying the possible parses for each sentence in the corpus, as well as calculating the parameters for the production probabilities in the grammar. We use the inside-outside algorithm to integrate over all possible parses and find the set of production probability parameters that maximize the likelihood of the grammar on the observed data (Manning and Schütze 1999; Johnson 2006). We evaluate Equation 3 in the same way, using the maximum-likelihood parameter values but integrating over all possible parses of the corpus. Sentences with longer derivations will tend to be less probable, because each production that is used contributes a multiplicative factor that is less than one. This notion of simplicity in derivation captures an inductive bias that favors sentences with more economical derivations.

3. Results

We first compare the three grammars R-CFG, B-CFG, and N-CFG, which differ from each other only in the recursive nature of some rules. As Figure 1 shows, the grammar with the highest posterior probability is B-CFG, which contains both non-recursive and recursive rules. It is neither the simplest grammar nor the one with the best fit, but offers the best tradeoff overall. As expected, the grammar with recursive rules (R-CFG) is the simplest, but the gain in prior probability is balanced by a loss in the likelihood caused by overgeneralization due to (relatively high-probability) recursive rules.

[Figure 1 goes here]

B-CFG contains recursive rules, but those rules have a small probability of expansion thanks to the non-recursive rules, which “absorb” much of the probability mass. Thus, B-CFG

does not allocate as much probability mass to non-observed  recursively-generated sentences and therefore over-generalizes less. The depth-limited grammar  over-generalizes least of all since it lacks recursion entirely - but, as Chomsky predicted, such a grammar is more complex than the others, and it is penalized here on that basis.

How might we interpret the preference for a grammar that contains recursive rules but also their non-recursive counterparts? Perhaps syntax, while fundamentally recursive, could usefully employ non-recursive rules to parse simpler sentences that recursive rules could parse in principle. This would not change the expressive capability of the grammar, but might dramatically decrease the cost of recursion. Indeed, if adult grammars contain both recursive and non-recursive rules, this (in addition to performance considerations) might partially explain the empirical lack of multiply-center-embedded sentences.

Is the role of recursion different in subject and object noun phrases? There are several reasons to think it might be: people have much less difficulty processing sentences with recursive object NPs, and the corpus contains many more recursively-generated object NPs than subject NPs. (For instance, the vast majority of subject NPs are pronouns or proper nouns, as in the sentences *He fell* or *Bambi is home*, whereas there are many more complex object NPs, as in the sentence *Butch is a little boy who plays with Bozo the clown*). Might the “shadow” recursive rules provide a benefit, then, only in the case of subject NPs? To test this we compare the grammars with distinct subject and object noun phrases. The results are shown in Figure 2.

[Figure 2 goes here]

Several things are notable about these results. Interestingly, all of the grammars with distinct subject and object NPs have higher probability than any of the grammars without (in Figure 1). This supports the intuition that subject and object NPs have distinct properties. Also, as hypothesized, the “shadow” non-recursive elements provide a benefit when they occur in subject NPs, but not in object NPs. Indeed, grammars with only recursive NP_o rules outperform grammars with both recursive and non-recursive NP_o rules, which in turn outperform grammars without any recursive NP_o rules at all. The story is different with subject noun phrases: there are so few recursive expansions of subject noun phrases in this corpus that grammars with only recursive productions perform extremely poorly. The depth-limited grammars have a somewhat higher probability, but (as before) do not fit the data well enough to compensate fully for their lower prior probability. The best-performing grammar of all, then, contains both recursive and non-recursive rules in subject NPs, but only recursive rules in object NPs, where the cost of recursion is far less.

However, these results depend in an interesting way on the data. Because it is child-directed speech, the corpus contains many sentence fragments, including bare NPs. The results in Figure 2 occur when the grammar assumes that bare NPs are instances of subject noun phrases (a sensible choice, given the frequency and obligatoriness of subjects in English). If instead the sentence fragments are analyzed as instances of object noun phrases, the most preferred grammars contain depth-limited rules (i.e., no recursion) in the subject NP. This is reasonable since under such an analysis, none of the subject NPs in the corpus are of depth 2 or greater; a grammar with *any* recursive subject NP rules is therefore penalized in the likelihood while providing no benefit in prior probability. This suggests that the benefit of recursive rules occurs only if there is a high enough depth of embedding in the input. It also illustrates the sensitivity of

our approach to the precise nature of the data, which is both a benefit and a limitation: a rational learner *should* be sensitive to the data, and our model is sensitive in reasonable and easily interpretable ways. However, it does underline the need for caution in interpreting these results: further investigation on a much wider range of corpora is necessary in order to draw strong conclusions about the nature of recursive rules in adult grammars of English.

5. Discussion

This Bayesian framework for grammar induction can be used to quantitatively evaluate grammars with respect to one another (and a corpus of data) by calculating an ideal tradeoff between simplicity and goodness-of-fit. We employ this framework to evaluate the role of recursion, addressing whether a grammar with recursive noun phrases would be preferred by a rational learner even if the input - a corpus of typical child-directed speech - does not contain sentences with many levels of embedding. Our results suggest that depending on the distribution of recursively-generated sentences in the input, grammars that contain both recursive and corresponding non-recursive “shadow” elements may be most probable: these grammars maximize the tradeoff between simplicity (which favors recursive elements) and measures of goodness-of-fit (which generally do not).

We are not claiming that adult grammars of English necessarily do contain shadow non-recursive rules. All of the grammars evaluated here are oversimplified in many ways, and our corpus consists of child-directed rather than adult-directed speech. Further work on a wider variety of corpora will therefore be necessary to draw any definite conclusions. Our results do suggest that under some circumstances it may be rational to generalize beyond the input. In particular, we demonstrate that shadow rules may improve a grammar's goodness-of-fit while allowing it to retain the expressive power (and some of the simplicity) of recursion. This suggests that a human learner might rationally conclude that particular rules have recursive versions given only finite, and realistic, input. That is, the knowledge that grammar can contain recursive rules *need not* be innate (although it could be, and a learner must be able to represent recursion so as to consider it as a hypothesis).



In principle, future work based on these ideas could help to settle the debate over recursion in Pirahã. One would need a representative corpus of Pirahã speech as well as (at least) two competing grammars of Pirahã: one recursive and one non-recursive – as well as possibly others populating the representational middle ground we have explored here. Given this, an analysis directly analogous to the one we have pursued here would provide an objective basis for addressing this issue.

This approach is valuable because it yields a rational, ideal standard by which it may be possible to judge human behavior, not because human learning is necessarily ideal. Bayesian methods offer what is arguably the best general way to formalize rational inductive inference (Jaynes 2003), and are consistent with an information theoretic perspective (Li and Vitányi 1997). Rational models of learning and inference based on Bayesian and information-theoretic statistical principles have been useful for understanding many aspects of human cognition (Anderson 1991; Chater and Oaksford 1999), including language (Chater and Manning 2006; Chater and Vitányi 2006; Dowman 2000). Even if human beings ultimately do not behave as a rational learner would, being able to establish how an optimal learner would perform provides an important means of evaluating how humans actually *do* perform, as it has here.

6. Acknowledgments

We would like to thank Tim O'Donnell, Michael Frank, and three anonymous reviewers for helpful feedback. This work was supported by an NDSEG graduate fellowship (AP), an NSF graduate fellowship (AP), the Paul E. Newton Career Development Chair (JBT) and the James S. McDonnell Foundation Causal Learning Collaborative Initiative (JBT).

Notes

¹ When we speak of evaluating a grammar, we presume unless stated otherwise that it is evaluated with respect to a finite corpus of sentences of the language, and relative to all of the (infinitely many) grammars that can generate the sentences in that corpus. Grammars that cannot generate the sentences may be eliminated on that basis. The corpus is assumed to contain positive evidence only.

² See Perfors, Tenenbaum, and Regier (under review) for a more detailed explanation of the corpus and the computational model, which are identical to the one used here.

³ <http://www.psychology.adelaide.edu.au/personalpages/staff/amyperfors/research/lingrevrecursion/> contains the full grammars and corpora.

⁴ Parts of speech used included determiners (*det*), nouns (*n*), adjectives (*adj*), comments like “mmhm” (*c*), prepositions (*prep*), pronouns (*pro*), proper nouns (*prop*), infinitives (*to*), participles (*part*), infinitive verbs (*inf*), conjugated verbs (*v*), auxiliary verbs (*aux*), complementizers (*comp*), and wh-question words (*wh*). Adverbs and negations were removed from all sentences. Additionally, whenever the word *what* occurred in place of another syntactic category (as in a sentence like *He liked what?*), the original syntactic category was used; this was necessary in order to simplify the analysis of all grammar types, and was only done when the syntactic category was obvious from the sentence.

⁵ Removed types included topicalized sentences (66 individual utterances), sentences containing subordinate phrases (845), sentential complements (1636), conjunctions (634), serial verb constructions (460), and ungrammatical sentences (443).

References

- Anderson, John
1991 The adaptive nature of human categorization. *Psychology Review* 98 (3): 409-429.
- Bach, Emmon, Colin Brown, and William Marslen-Wilson
1986 Crossed and nested dependencies in German and Dutch: A psycholinguistic study. *Language and Cognitive Processes* 1: 249-262.
- Brown, Roger
1973 *A first language: The early stages*. Cambridge, MA: Harvard University Press.
- Chater, Nick, and Chris Manning
2006 Probabilistic models of language processing and acquisition. *Trends in Cognitive Sciences* 10 (7): 335-344.
- Chater, Nick, and Mike Oaksford
1999 Ten years of the rational analysis of cognition. *Trends in Cognitive Sciences* 3: 57-65.
- Chater, Nick, and Paul Vitányi
2006 'Ideal learning' of natural language: Positive results about learning from positive evidence. *Journal of Mathematical Psychology* 51 (3): 135-163.
- Chomsky, Noam
1956 Three models for the description of language. *IRE Transactions on Information Theory* 2: 113-123.
1957 *Syntactic structures*. The Hague: Mouton.
1959 On certain formal properties of grammars. *Information and Control* 2: 137-167.
1965 *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.
- Christiansen, Morten, and Nick Chater
1999 Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science* 23(2): 157-205.
- Dowman, Mike
2000 Addressing the learnability of verb subcategorizations with Bayesian inference. *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*.
- Everett, Daniel
2005 Cultural constraints on grammar and cognition in Pirahã: Another look at the design features of human language. *Current Anthropology* 46(4): 621-646.
2007 Cultural constraints on grammar in Pirahã: A reply to Nevins, Pesetsky, and Rodrigues (2007). <http://ling.auf.net/lingbuzz/000427>
- Fitch, William Tecumseh, and Marc Hauser
2004 Computational constraints on syntactic processing in a nonhuman primate. *Science* 303: 377-380.
- Gentner, Timothy, Kimberly Fenn, Daniel Margoliash, and Howard Nusbaum
2006 Recursive syntactic pattern learning by songbirds. *Nature* 440: 1204-1207.
- Gibson, Edward
1998 Linguistic complexity: locality of semantic dependencies. *Cognition* 68:1-76.

- Gold, E. Mark
1967 Language identification in the limit. *Information and Control* 10(5): 447-474.
- Goldwater, Sharon, Thomas Griffiths, and Mark Johnson
2006 Interpolating between types and tokens by estimating power law generators. *Neural Information Processing Systems* 18.
- Hauser, Marc, Noam Chomsky, and William Tecumseh Fitch
2002 The faculty of language: What is it, who has it, and how did it evolve? *Science* 298 (5598): 1569-1579.
- Horning, James Jay
1969 A study of grammatical inference (Tech. Rep. No. 139). Stanford University.
- Jaynes, Edwin
2003 *Probability theory: The logic of science*. Cambridge: Cambridge University Press.
- Johnson, Mark
2006 Inside-outside algorithm. <http://www.cog.brown.edu/~mj/code/inside-outside.tgz>
- Jurafsky, Daniel, and James Martin.
2000 *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall.
- Li, Ming, and Paul Vitányi
1997 *An introduction to Kolmogorov complexity and its applications*. New York: Springer Verlag.
- MacWhinney, Brian
2000 *The CHILDES project: Tools for analyzing talk* (3rd ed.). Lawrence Erlbaum Associates.
- Manning, Chris, and Heinrich Schütze
1999 *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.
- Marks, Lawrence
1968 Scaling of grammaticalness of self-embedded English sentences. *Journal of Verbal Learning and Verbal Behavior* 5: 965-967.
- Miller, George, and Noam Chomsky
1963 Finitary models of language users. In *Handbook of Mathematical Psychology*, Vol. 2, R. Duncan Luce, Robert Bush, and Eugene Galanter (eds.), 419-492. New York: Wiley.
- Nevins, Andrew, David Pesetsky, and Cilene Rodrigues
2007 Pirahã exceptionality: A reassessment. <http://ling.auf.net/lingBuzz/000411>
- Perfors, Amy, Joshua B. Tenenbaum, and Terry Regier
under review The learnability of abstract syntactic principles. *Cognition*.
- Pinker, Steven, and Ray Jackendoff
2005 The faculty of language: What's special about it? *Cognition* 95: 201-236.
- Solomonoff, Ray
1978 Complexity-based induction systems: Comparisons and convergence theorems. *IEEE Transactions on Information Theory* 24: 422-432.

Weckerly, Jill, and Jeffrey L. Elman

1992 A PDP approach to processing center-embedded sentences. *Proceedings of the 14th Annual Conference of the Cognitive Science Society*, 139-156.

Zipf, George

1932 *Selective studies and the principle of relative frequency in language*. Cambridge, MA: Harvard University Press.

Table 1

Sample rules from each of the three main grammars. These are chosen to illustrate the differences between each grammar, and may not be an exhaustive list of all the expansions of any given non-terminal.

Grammar with recursive rules (R-CFG)
$\text{NP} \rightarrow \text{NP PP} \mid \text{NP CP} \mid \text{NP C} \mid \text{N} \mid \text{det N} \mid \text{adj N} \mid \text{pro} \mid \text{prop}$
$\text{N} \rightarrow \text{n} \mid \text{adj N}$
Grammar with both recursive and non-recursive rules (B-CFG)
$\text{NP} \rightarrow \text{NP PP} \mid \text{NP CP} \mid \text{NP C} \mid \text{NN PP} \mid \text{NN CP} \mid \text{NN C} \mid \text{NN}$
$\text{NN} \rightarrow \text{det N} \mid \text{N} \mid \text{adj N} \mid \text{pro} \mid \text{prop}$
$\text{N} \rightarrow \text{n} \mid \text{adj N}$
Grammar with non-recursive (depth-limited) rules (N-CFG)
$\text{NP} \rightarrow \text{N2 PP} \mid \text{N2 CP} \mid \text{N2 C} \mid \text{NN P} \mid \text{NN CP} \mid \text{NN C} \mid \text{NN}$
$\text{N2} \rightarrow \text{NN PP} \mid \text{NN CP} \mid \text{NN C} \mid \text{NN}$
$\text{NN} \rightarrow \text{det N} \mid \text{N} \mid \text{adj N} \mid \text{pro} \mid \text{prop}$
$\text{N} \rightarrow \text{n} \mid \text{adj N}$

Figure 1

Log posterior probability of three grammars with three different types of NP rules. The grammar with both recursive and non-recursive rules (B-CFG) has higher probability than either the depth-limited grammar (N-CFG) or the fully recursive grammar (R-CFG). Note that because log probability is negative, smaller absolute values correspond to higher probability. If two grammars have log probabilities that differ by n , their actual probabilities differ by e^n . The grammar with the highest probability (B-CFG) is thus e^{63} times more probable than the grammar with the next highest (N-CFG). Note: log priors = -1394 (B-CFG), -1471 (N-CFG), -1085 (R-CFG); log likelihoods = -25827 (B-CFG), -25813 (N-CFG), -26375 (R-CFG).

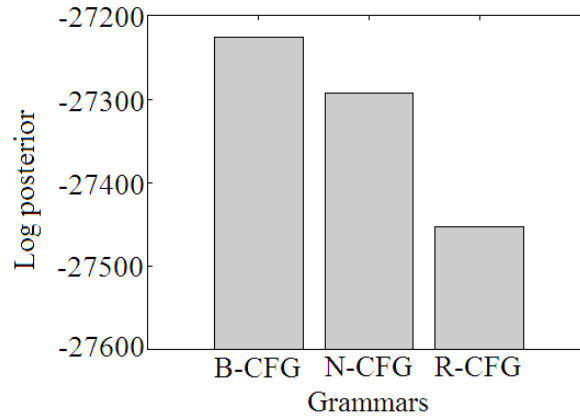


Figure 2

Log posterior probability of the nine grammars that systematically vary how recursion in NP_S and NP_O is represented. As before, because log probability is negative, smaller absolute values correspond to higher probability. Overall, the most probable grammar is the one whose subject NPs contain both recursive and non-recursive rules (B-CFG), but whose object NPs contain only recursive rules (R-CFG).

