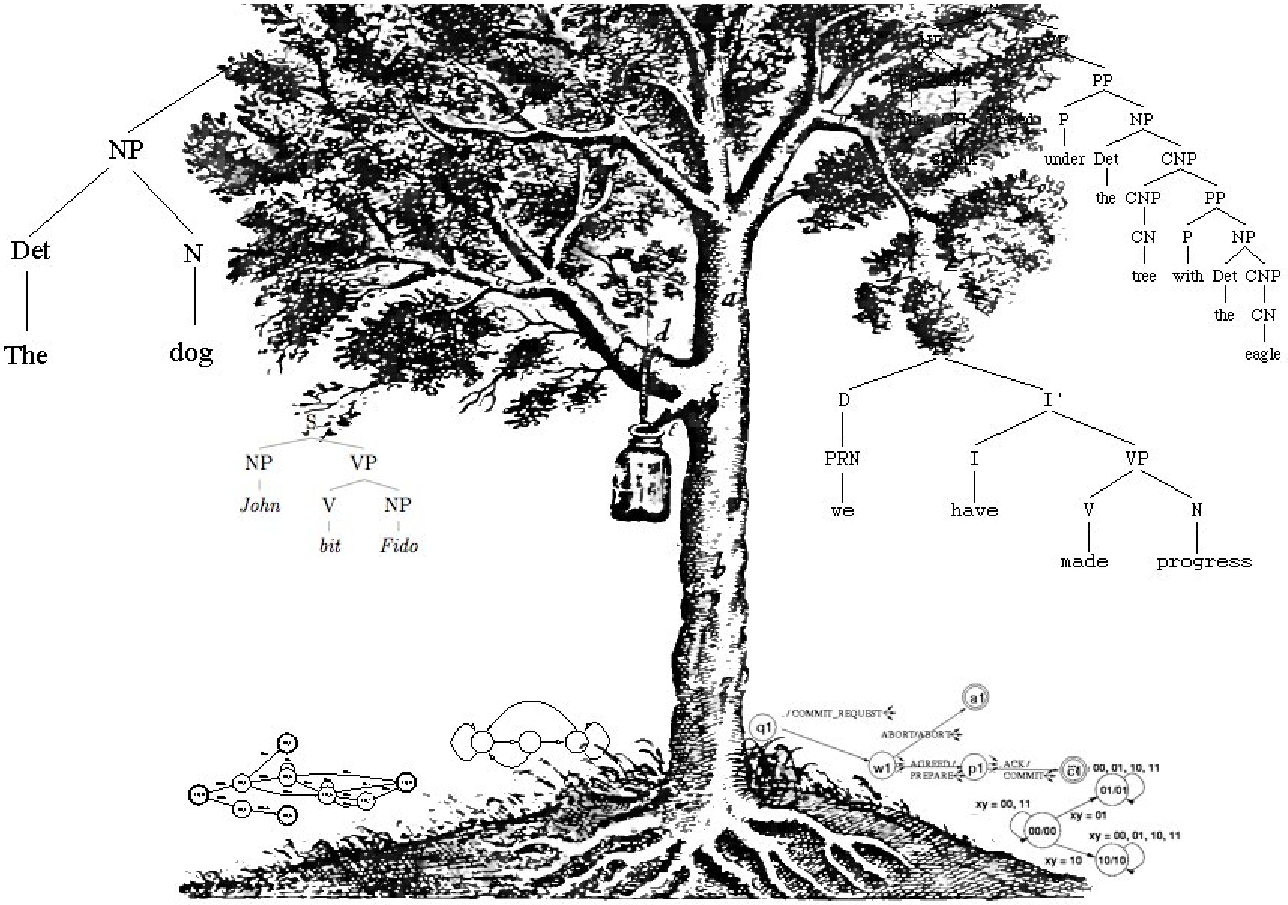
# Grammar as a Choice

## Inferring Mechanisms of Structured Learning in a Serial Choice Reaction Time Task



*Draft 2nd Year Project Proposal – Oct 6, 2008*

Jamie Alexandre (jdalexan@cogsci.ucsd.edu)
Department of Cognitive Science, UCSD

## ABSTRACT

The process of learning a human language involves the inference of large numbers of statistical and structural regularities from a finite history of linguistic input. The mechanisms underlying this process of induction, their dependence on genetic vs experiential factors, and their specificity to language have been hotly debated for decades. Syntax, in particular, has long been defended as a bastion of empirical unlearnability, necessitating a highly constrained genetic endowment as the basis for language acquisition. The artificial grammar learning paradigm can serve as a useful tool for studying the process of syntactic acquisition, and as a testing bed for computational models of grammar learning. This paper presents an approach based on a serial choice reaction time task, which provides incremental reaction time data that can be quantitatively compared with the predictions of probabilistic incremental expectancy models. The concept of surprisal (Hale, 2001; Levy, 2008) is explored as a means of mapping between the RT data and the predictions of the probabilistic models.

## INTRODUCTION

Language is rich with structure, and decoding this structure is an important part of comprehending any linguistic utterance. The question, then, is what computational mechanisms underlie the processing of structured sequences? Even the mechanisms underlying the verbatim encoding of serial order in working memory remain largely mysterious, let alone the processes that allow for more abstract representations of a sequence's underlying structure.

This paper focuses in particular on the question of syntax, which has seen a long and rich history of debate in both scholarly and popular circles, dating back at least as far as Dionysius Thrax's outline of Greek syntax over two millennia ago (Jurafsky & Martin, 2008). More recently, syntax has played a pivotal role in disputes over the innateness of language, with Chomsky's universal grammar forming the cornerstone of his case for a genetically determined "language acquisition device" (Chomsky, 1965).

The acquisition of syntax can be framed as a problem of inferring structure from a linear sequence. The nature of this "structure," however, is a primary point of contention. An n-gram language model is perhaps the simplest representation of sequential dependency; it posits very local, statistical relationships between lexical items in a sequence, such that the probability of a particular item occurring is a function of the *n - 1* items that preceded it. N-gram models, despite their many shortcomings, have proven to be incredibly powerful tools for many tasks, both practical and theoretical. At the other end of the spectrum, we have systems based on phrase structure grammars, which view linguistic sequences as the product of rule-based derivational processes (we will examine such systems in more detail below). A key feature of such grammar-based systems is the potential for long-range dependencies, such that the appropriateness of a particular item can depend on what occurred at an arbitrarily distant point earlier in the sequence. A third proposal, notable for the biological plausibility of its underlying mechanism as well as the graded nature of its decay in performance for dependencies of increasing sequential distance, comes from the connectionist literature. Elman's (1990) Simple Recurrent Network (SRN) model is a standard three-layer feed-forward network, with the addition of a context layer that maintains a copy of the hidden layer's state from the previous timestep, which then feeds back into the hidden layer via trainable weights. This allows an SRN, trained on sequential data, to use its hidden layer representations to maintain *relevant* contextual information over theoretically unbounded (though for most practical purposes, quickly decaying) distances.

## FORMAL LANGUAGES, GRAMMARS, AND AUTOMATA

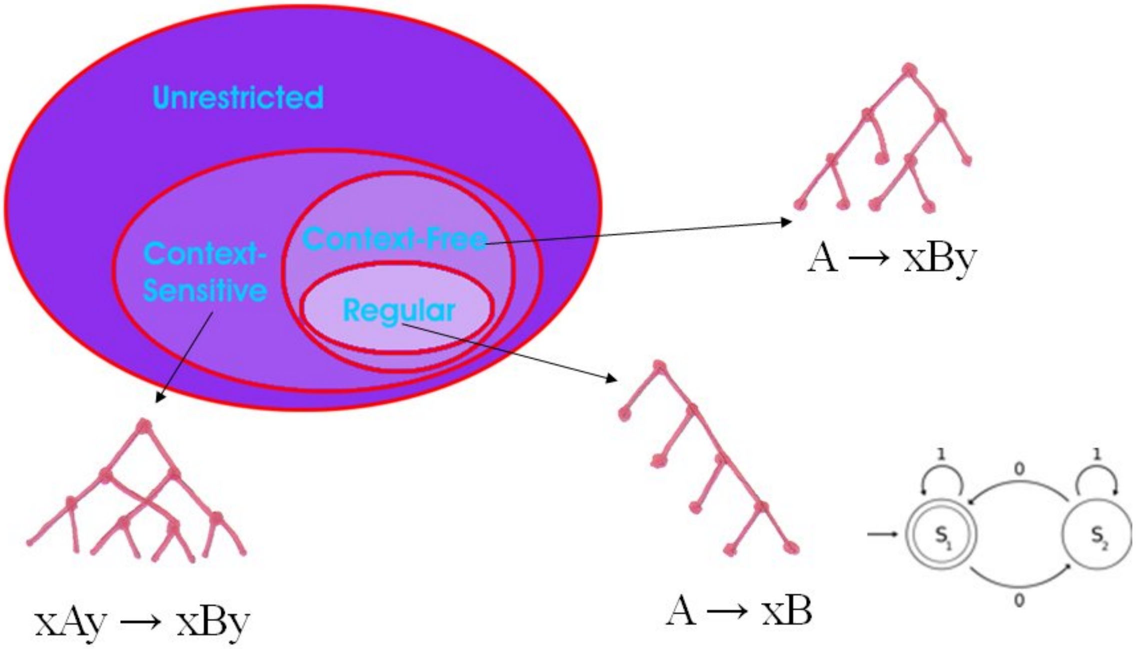
This section will briefly outline the elements of formal language theory, as this will be useful for the discussion to follow.

Formal language theory defines a *language* as a set of ordered sequences ("strings" or "sentences") consisting of tokens from an *alphabet* (a set of symbols, words, or other atomic items). A sentence, then, belongs to the language (or, is *grammatical* in the language) if and only if it is a member of the set of strings that constitutes the language. This set can be specified in various ways (e.g. verbally – "the set of all palindromes over alphabet {a,b}"), but in the end it is just a (potentially infinite) set of strings.

A formal grammar is a set of production rules (such as S → A B) that transform one string into another through a series of replacements. In addition to the *terminal symbol* set that consists of the words or characters in the alphabet, a grammar also posits a set of *variables* or *non-terminal symbols*, which play a role in the derivation but do not show up in the final string. Beginning with a "start" state, the successive application of production rules eventually generates the final string. A *syntax tree* can be generated in tandem with this derivational process by initializing a tree with the start state as the root label, and then expanding a node when the corresponding non-terminal symbol is replaced, adding the symbols on the right-hand side of the rule as the node's children.

## THE CHOMSKY HIERARCHY

Not all languages were created equal, and the set of formal languages can in fact be arranged into a containment hierarchy, with each successive subset having increasingly constrained production rules. In general, the more highly constrained the production rules, the simpler the corresponding grammar. Simpler grammars also correspond to simpler languages, and can be implemented by simpler classes of automata.



*The Chomsky hierarchy of formal languages*

The two of most interest for our present purposes are regular and context-free grammars, as these are the grammars typically used in modeling human languages. Regular grammars are the simplest, containing only right-branching (or left-branching) derivations, and can be processed by finite state machines (bottom right). Context-free grammars can contain both right and left-branching derivations (top right), and require automata that have memory, in the form of a pushdown stack. Context-free grammars are thought to be sufficiently powerful to model most of the structure found in natural languages, although some languages (most notably Dutch and Swiss-German) contain structures with cross-dependencies that have been said to require context-sensitive grammars.

## Surprisal

Surprisal, or self-information, is a notion from information theory that quantifies the amount of novel information that a particular event carries with it. The occurrence of a very probable event doesn't tell us very much about the world, and thus has lower self-information than an unexpected (or *surprising*) event. An event's surprisal is defined as its negative log probability (if the logarithm has base 2, then the surprisal value is measured in bits):

$$-\log\left(P(x)\right)$$

The concept of surprisal has been used in psycholinguistics as a potential measure of incremental processing difficulty, and is thus expected to correlate with behavioral measures such as reading times in eye-tracking studies, and response times in self-paced reading studies (Hale, 2001; Levy, 2008).

The surprisal model requires that we adopt some measure of the probability of a word's occurrence given the preceding sentential context. As discussed above, many candidate models of conditional word probability exist, and it is not immediately clear which one is ideal for the calculation of surprisal. Hale (2001) uses a probabilistic Earley parsing algorithm (discussed below) to generate incremental word probabilities, using the resulting surprisal values to explain the garden path effect. Levy (2008) uses a similar model to explain a wide-range of effects found in other psycholinguistic theories such as predictability (e.g. effect of Cloze probability), locality theory (e.g. preference for local dependencies), competition/dynamical models (e.g. greater ease in highly constrained contexts), the tuning hypothesis (e.g. effect of structural frequency), and connectionist models (e.g. predictions made by an SRN). The case of the SRN is particularly interesting, because there are significant divergences between the predictions made by an SRN and a PCFG-based surprisal model, particularly for constructions such as recursive center-embeddings, which PCFGs process flawlessly, and SRNs – much like humans – have difficulty processing beyond a few levels of embedding (Christiansen & Chater, 1999). On the other hand, however, Frank (2008) tested a surprisal model against human data, comparing PCFG- with SRN-generated probabilities, and found that the SRN produced more accurate probabilities, but that the PCFG produced probabilities that better matched the human data. He concludes from this, firstly, that subjective probabilities do not necessarily coincide with objective probabilities, and secondly, that the PCFG may in fact be a better model of human performance. Other surprisal studies have used n-gram statistics, such as a trigram model with Kneser-Ney smoothing (Smith & Levy, 2008), and also shown close correspondences with human data. The debate thus continues, but the general lesson here is that surprisal can be a useful tool in investigating the strategies and representations at play in human language comprehension.

## INCREMENTAL PROBABILISTIC PARSING

A stochastic/probabilistic context-free grammar (PCFG) is a CFG with the added feature that each production rule is associated with a probability. The choice of which rule to apply, when expanding a non-terminal symbol, is decided according to the probability distribution over the rules that have that symbol on their left-hand side. Parsing using PCFGs helps to avoid some of the obstacles faced by standard CFG parsing, particularly in dealing with structurally ambiguous inputs, as a PCFG assigns a probability to each potential parse tree (the product of the probabilities of all the rule applications in its derivation), and the tree with the highest probability will (we would hope) tend to be the correct one.

Given a PCFG as a model of linguistic structure, we can also compute a number of other pieces of information that are useful in studying incremental language processing. I will present these in list-form, as they build in a stepwise fashion upon one another:

- The probability of a string is the sum of the probabilities of all its parse trees.
- The probability of a string prefix is a sum over the probabilities of all possible completions of the prefix.

- The probability that a particular symbol $w_i$ will appear following the string prefix $w_1..w_{i-1}$ can be computed by dividing the probability of the prefix with that symbol appended, $P(w_1..w_i)$, by the probability of the prefix, $P(w_1..w_{i-1})$

The above calculations are due to Jelinek & Lafferty (1991), who developed a modification to the CYK parsing algorithm that efficiently computes the above quantities, and are further discussed in Stolcke (1995), who modified the Earley algorithm to do the same.[1]

Note that up to this point we are assuming that we already know the underlying PCFG. All that a language learner has to work with, however, is a finite history of experience with surface linguistic forms (the sentences) generated by the grammar. The structure that a learner induces from this input – particularly when the quantity of input is limited, relative to the grammar's complexity – will not necessarily coincide with the grammar that produced the sentences, as long as the two are roughly compatible with respect to the input history. Hence, I would add an additional step to the list above:

- The probability of a string, given a history of input strings, is the sum of the probabilities of that string for each of the grammars inferable from the input history, weighted by the probabilities of each of those possible grammars:

  $$P(w_1..w_n|History) = \sum_{G_i \in G} P(w_1..w_n|G_i)\, P(G_i|History)$$

  where *$w_1..w_n$* is the string in question, *History* is the individual's complete history of input, and *G* is the set of all possible grammars.

This summing out of grammars emphasizes even more explicitly the fact that what is relevant to us here about PCFG-based representations is the probabilistic models they generate, without commitment to the psychological reality of parse trees, production rules, or similar mental constructs. It's possible for a grammar-based system to be predictive without being descriptive.

Factoring in an individual's history of input also emphasizes *development*; we can now look at how one's expectations change over the course of learning, gaining insight into the rate at which a system is acquired, as well as possible shifts in strategy over the course of learning.

[1] In the current work, I have built upon Roger Levy's implementation of Stolcke's incremental parsing algorithm, available at: http://idiom.ucsd.edu/~rlevy/prefixprobabilityparser.html

## PALINDROMES: AN EXAMPLE OF INCREMENTAL EARLEY IN ACTION

Palindromes are a canonical example of a context-free language, often defined as:

$$\{ww^R : w \in \Sigma^+\}$$

An example of this, on the alphabet Σ={r, g, b, y}, would be “r y g b y y b g y r” – the characteristic feature being the symmetry around the center point. An example of a PCFG that generates strings from this language (avoiding a simpler version that uses lambda productions) would be:

| | |
|---|---|
| 0.2 | S→ rSr |
| 0.2 | S→ gSg |
| 0.2 | S→ bSb |
| 0.2 | S→ ySy |
| 0.2 | S→ E |
| 0.25 | E → rr |
| 0.25 | E → gg |
| 0.25 | E → bb |
| 0.25 | E → yy |

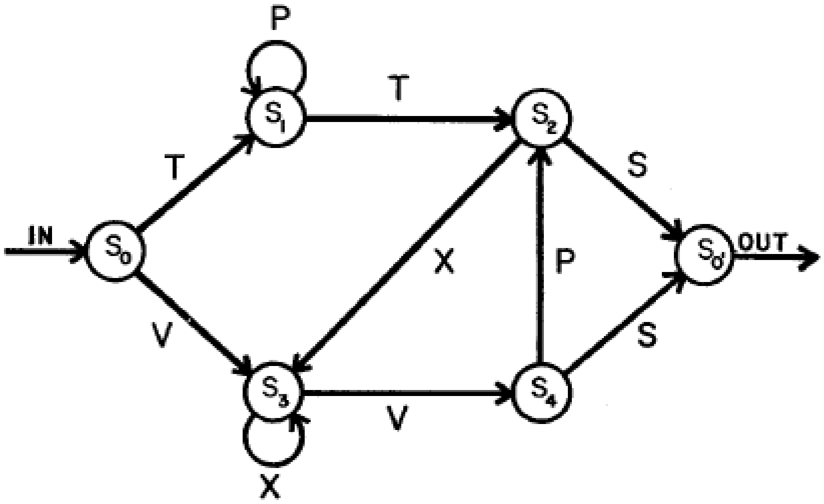
This grammar can be passed to the probabilistic Earley parser, which can then serve as a model of incremental expectancies, producing a distribution over possible continuing symbols, given a prefix. Based on the symbol that actually does follow, we can then calculate its surprisal value by taking the negative log probability of that symbol having occurred. In this way, we can calculate surprisal values for every word in a sequence, such as the following example:

| **Char #:** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Char:** | b | g | b | y | r | r | y | y | r | r | y | b | g | y |
| **Surprisal** | 2.0 | 2.32 | 2.32 | 2.32 | 2.32 | 1.32 | 0.74 | 3.91 | 0.74 | 0.15 | 0.04 | 0.07 | 0.002 | 13.93 |
| **Probab.** | 0.25 | 0.2 | 0.2 | 0.2 | 0.2 | 0.4 | 0.6 | 0.07 | 0.6 | 0.9 | 0.97 | 0.96 | 0.99 | 6.0E-5 |

There are several very interesting things to note here. Firstly, the surprisal at the beginning (2.0) is indicative of the fact that each of the 4 symbols ($2^2$) is equally likely. Symbols 2-5 have a probability of 1/5, because there is a 2/5 chance (see word 6) that the previous symbol will be repeated, since the middle of the string always involves such a repetition. Words 6 and 7 are examples of a garden path, as they begin to mirror the preceding symbols, and thus surprisal begins to decrease (the model “thinks it knows” where this is going). Word 8, however, is a rude awakening, as the mirroring pattern is broken, indicating that we still have a ways to go. Things start looking good, probabilities increase, and surprisal decreases, from words 9-13, as it becomes almost certain that we’re on the path home, which leads to extremely high surprisal when it becomes clear, on word 14, that we’re not even halfway there yet.

## ARTIFICIAL GRAMMAR STUDIES

Artificial grammar research began with Reber's (1967) work on the implicit learning of structured stimuli, and much of the work since then has followed in a similar vein. Reber's experiment was described to subjects as a memorization task, and subjects were required to reproduce strings of letters (generated either by a simple finite state machine or randomly) that were printed on cue cards. The primary finding was that, as expected if the subjects had learned "the lawfulness that existed in the stimulus array," reproduction errors over successive blocks of presentation decreased more rapidly for the strings generated by the grammar than for the randomly generated strings. Reber used the following finite state machine to generate the grammatical strings:



There are several things worth noting about this grammar, particularly as it has served as the template for many of the grammars used by generations of artificial grammar experiments (including recent ones), becoming known as a "Reber machine". The more-or-less monotonic progression from left to right (with the exception of a single back-transition from $S_2$ to $S_3$) leads to the production of highly predictable strings – note, even, that grammatical strings *always* end with an "S". While this simple system served Reber's purpose – to demonstrate acquired sensitivity to statistical regularity without explicit learning – it tells us very little about the nature of *grammar* induction, which makes it surprising that very similar (sometimes identical) Reber machines became the gold standard within the artificial grammar learning literature (e.g. Dienes et al, 1991; Baldwin & Kutas, 1997; Shanks et al, 1997; Carrión & Bly, 2007; Gebauer and Mackintosh, 2007).

Another important thing to note about Reber's experiment is the style of presentation; simultaneous display of all the symbols in a "sequence" allows for qualitatively different learning mechanisms than incremental exposure (enabling direct long-distance comparisons, for instance). The response measure, furthermore, is offline, which means that each string is treated as a single unit, preventing us from examining the role that particular substructures play in a subject's encoding of the grammatical regularity.

While many artificial grammar experiments continued in the same vein as Reber's original work, many variants have also been explored. Some researchers have tried to bridge the gap between the abstractness of artificial grammar learning and the rich input that accompanies natural language acquisition by integrating semantic content into the artificial language model; for instance, Friederici et al (2002) construct a language around object names and actions within a mini-world presented to the subject, such that the "meaning" of sequences is learnt in addition to the sequence structure. Other researchers have tackled the issue of recursive embedding; Christiansen & Chater (1999) found that SRN performance on center-embedded structures closely matched human performance in a grammar learning task that used the same materials. More recently, brain imaging tools such as EEG and fMRI have been used to localize artificial grammar processes both temporally and spatially, revealing insights into the relationship between the mechanisms underlying the processing of artificial grammars and the mechanisms that allow for natural language comprehension (e.g. Baldwin & Kutas, 1997; Hoen & Dominey, 2000; Friederici et al, 2002; Opitz & Friederici, 2003; Lieberman et al, 2004; Carrion & Bly, 2007; Christiansen et al, 2007).

## SERIAL CHOICE REACTION TIME TASK

The goal of the present study is to obtain estimates of a subject's online string continuation expectancies, so that these may be compared with the predictions made by a variety of language models trained on the same input history as the subject. The traditional measures of successful acquisition in artificial grammar experiments – such as grammaticality judgments, or error rates in recall – are not able to provide the incremental (symbol-by-symbol) expectancy data that we require. Having the subject perform prefix completion (i.e. explicit next-symbol prediction) can be very useful in this context, but this results in very sparse data, and disrupts the flow of presentation. Self-paced reading brings us one step closer to what we want, as it provides per-word response times that have been shown, at least in natural language contexts, to correlate highly with a word's degree of unexpectedness. However, effects in self-paced reading times are very susceptible to a subject's degree of interest in the content being read, as it is very easy to rhythmically cycle through a sentence without paying any attention to it. The standard way to control for this is to include comprehension questions at the end of a set of sentences, but it is unclear how this could be extended to the case of an artificial grammar.

The experiments proposed here will use a technique employed by Cleeremans & McClelland (1991), known as a serial choice reaction task. Subjects respond to a stimulus (with one button per stimulus) by pressing the corresponding button as quickly as possible after perceiving stimulus onset. This method avoids several of the problems associated with self-paced reading; subjects must engage with the task in order to do well (Cleeremans & McClelland even provided financial rewards for speed and accuracy), and one would not expect to see the same sorts of RT spillover effects as are found in self-paced reading, as the stimulus must be processed before a button press can be initiated (since there are many buttons), localizing to some extent the effects of unexpectedness to the current symbol.

The use of surprisal in a serial choice reaction time task can also be motivated by the early experimental result that in a choice reaction time task with n equiprobable choices (buttons), reaction times are proportional to the log of the number of choices, a result known as *Hick's law* (Hick, 1952). This is simply a special case of surprisal, since:

$$\log(N) = -\log\left(\frac{1}{n}\right)$$

Furthermore, Crossman (1953) demonstrated that the relation holds even when some choices are more likely than others, in the context of a card sorting task. Interestingly, proper shuffling turned out to be a critical factor, as serial regularities led to marked increases in speed.

## Experimental Apparatus

Several potential interface devices have been acquired or assembled, and pilot testing will establish which of these is the most effective in terms of accurate RT measurement, subject comfort, and sensitivity of RTs to experimental manipulation. One option is to use a standard game controller, with 4 trigger buttons and additional thumb buttons, with stimuli presented on the computer screen. Keys on a computer keyboard could be used in a similar manner, as per Cleeremans & McClelland (1991). There are several reasons, however (including accuracy of RT measurement and proximity of the stimulus display to the response buttons) to use a custom-built device interfaced via a dedicated DAQ device.

A V-Tech Mini Wizard Handheld Game (originally produced in 1987) has been adapted to serve as a button box. This device is particularly suitable, as it consists of a sturdy box mounted with 4 large, colored buttons and a 3x3 array of LEDs to use for stimulus display. The buttons and LEDs are interfaced to the PC via a USB-powered LabJack U3 DAQ device, which has very high sampling rates and low command-response latencies, allowing for precision stimulus control and RT measurement via software on the PC.

The arrangement of the buttons relative to the LED array allows for a variety of stimulus-response mappings (see pictures below). In a simple reaction time variant, the row of 3 LEDs next to a button lights up to specify that button as the target. In a complex reaction time version of the task, mappings are learnt from arbitrary, symmetric LED patterns to particular buttons. Complex reaction times tend to be longer, and may involve processes that engage higher-level systems in ways that simple, perceptual reactions do not, and thus comparisons between the two styles could prove insightful.

A second potential button box is also being constructed, consisting of 8 thumb-sized pushbuttons (with short travel distance), arranged in a 2x4 array, with each button containing a separately controllable LED for use as stimulus cues. The box will be interfaced to the computer via the same LabJack device, providing the same advantages mentioned above in terms of measurement accuracy and flexibility.

## STIMULI

As the goal of the present study is to help determine the cognitive-computational mechanisms underlying the encoding of sequential structure, the stimuli will be designed to maximize diagnosticity in distinguishing between the various potential models.[2] Stimuli will be pseudo-randomly generated according to the following procedure:

1. Some initial small set of symbol strings are generated randomly.
2. Each of the candidate models is trained on the training set up to the current symbol.
3. The next symbol generated is the one for which the trained models assign maximally different continuation probabilities.
4. Stop if stimulus set is sufficiently long, otherwise goto step 2 and loop.

## METHODS

Subjects are told that they are participating in an experiment to test reaction times and attention. No mention will be made of the structured nature of the stimuli, as it has been found that making the task explicit decreases performance in cases where the structure is not highly salient (Cleeremans, 1993, p. 7).

Initial pilot testing will consist of generating pseudo-random sequences with fixed unigram probabilities, with the expectation that reaction times will be proportional to the negative log unigram probability for each button (according to the imbalanced variant of Hick's law, discussed above). This will (hopefully) help to confirm the following:

1. That the hardware is physically sensitive enough to capture accurate RTs.
2. That the hardware is an effective, intuitive interface device, avoiding excessive sources of RT noise.
3. That the learning paradigm is valid insofar as the RTs reflect a subject's expectancies, and the subject is at least able to learn the unigram probabilities.
4. That negative log probability is a reasonable choice for mapping between RTs and probabilities.

[2] Jeff Elman has recommended that it might be better to figure out what the key differences are between the models, and manually design stimuli to help distinguish them. While I wish to avoid making the sorts of assumptions that hand-constructed stimuli imply, it may be possible to first generate sets of maximally diagnostic stimuli as described here, and then analyze them to determine what the critical factors are, which could then be used in hand-constructing optimal, but better-controlled, stimuli.

## ANALYSIS

Previous serial choice reaction time experiments primarily focused their analyses on subjects' global learning curves, and the effect of inserting trials that violated the sequence's underlying pattern, in order to demonstrate that learning had taken place (beyond the effects of general practice at the task). Lewicki et al (1987), for instance, waited until the learning curve had leveled off, and then switched the rule underlying the sequence's predictability, noting from the resulting bump in response time that subjects had gained some sensitivity to the original rule and had been making use of this knowledge to reduce response times. Similarly, Cleeremans & McClelland (1991) interleaved a small proportion of ungrammatical sequences in with their stimuli set, noting that reaction times decreased more quickly and remained lower for the grammatical vs the ungrammatical sequences. But the data contains much more information than can be observed in the global learning curve – looking at exactly *where* in the structured data subjects show increased performance can give us hints as to what strategies they are using and the types of structural properties they are sensitive towards.

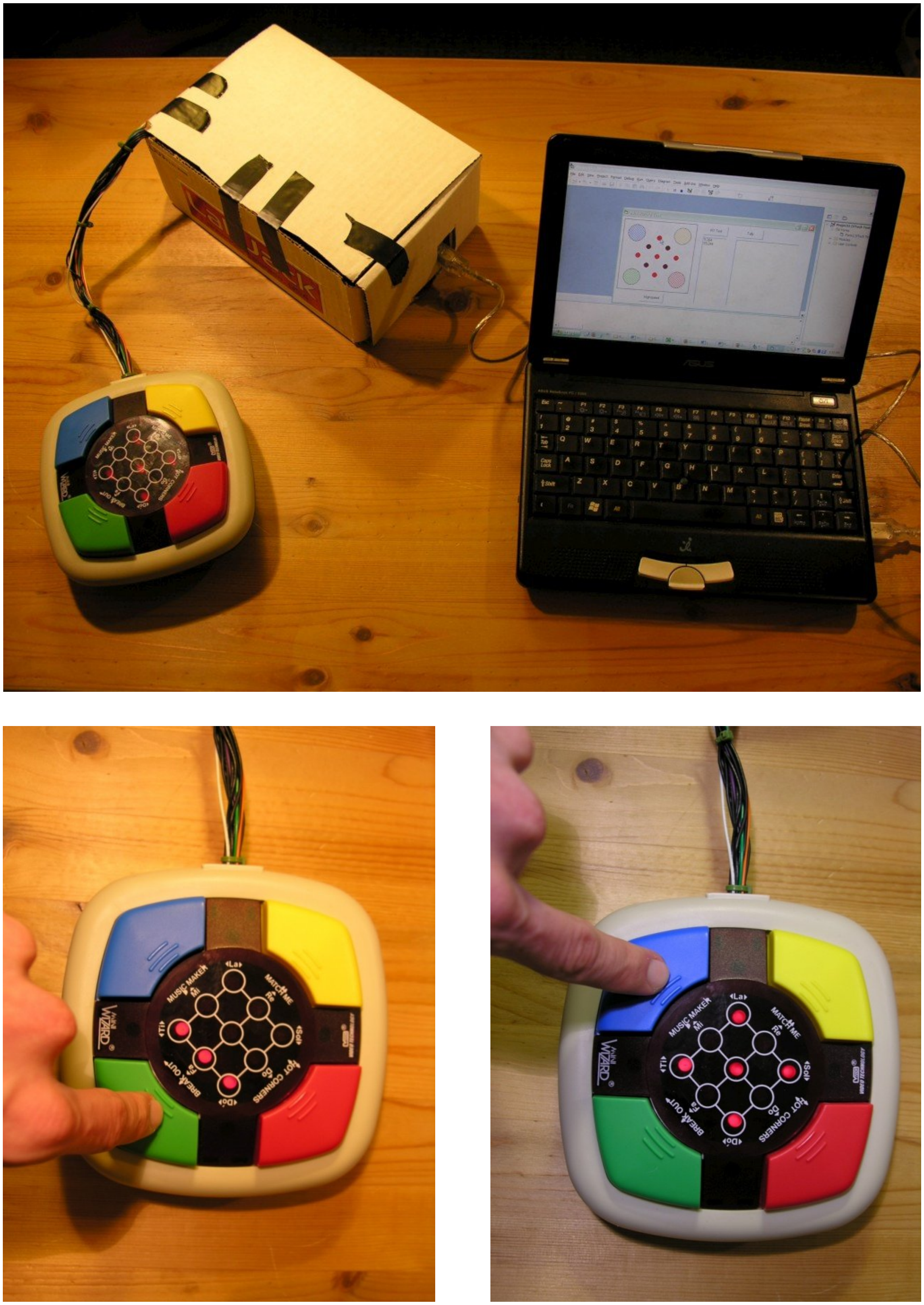
In the present study, the RTs obtained through the serial choice reaction time task will be interpreted as measures of a participant's subjective, incremental expectancies, which will then be compared with surprisal values generated by a variety of language models, to gauge the inductive strategies the subject may be using.

The language models to be tested will include:

- N-gram models for various N, with and without smoothing (e.g. Kneser-Ney)
- Finite state machine models
- Simple Recurrent Networks
- Incremental PCFG parsing using the probabilistic Earley algorithm

Of critical importance is the fact that each of these models will be trained on the precise input that a subject has been exposed to up to the current point in the experiment (rather than on a larger corpus). This allows us to obtain a picture of the *development* of the subject's and model's representations, rather than simply comparing fully trained systems.

It is not expected that any of these models will match the human RT data precisely, but rather that each will match it in different ways, and it is possible that it will be qualitative analysis of the particular points at which the models make different predictions from one another, and from the human data, that provide the greatest insight into what mechanisms may be at play in the processing of sequential structure.



(a) Working prototype of one experimental apparatus variant

(b) Example of simple reaction task

(c) Example of complex reaction task

# References

Baldwin, K. B. and Kutas, M. (1997). An ERP analysis of implicit structured sequence learning. *Psychophysiology*, 34(1):74-86.

Carrión, R. E. and Bly, B. M. (2007). Event-related potential markers of expectation violation in an artificial grammar learning task. *Neuroreport*, 18(2):191-195.

Chomsky, N. *Aspects of the Theory of Syntax*. MIT Press, 1965.

Christiansen, M. H. and Chater, N. (1999). Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science*, 23(2):157-205.

Christiansen, M.H., Conway, C., & Onnis, L. (2007). Neural Responses to Structural Incongruencies in Language and Statistical Learning Point to Similar Underlying Mechanisms. In *Proceedings of the 29th Annual Meeting of the Cognitive Science Society.*

Clark, A. (2001). Unsupervised induction of stochastic context-free grammars using distributional clustering. In *ConLL '01: Proceedings of the 2001 workshop on Computational Natural Language Learning*, pages 1-8, Morristown, NJ, USA. Association for Computational Linguistics.

Cleeremans, A. and McClelland, J. L. (1991). Learning the structure of event sequences. *Journal of experimental psychology. General*, 120(3):235-253.

Cleeremans, A. (1993). *Mechanisms of implicit learning: connectionist models of sequence processing*. MIT Press.

Dienes, Z., Broadbent, D., and Berry, D. (1991). Implicit and explicit knowledge bases in artificial grammar learning. *Journal of experimental psychology. Learning, memory, and cognition*, 17(5):875-887.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179-211.

Frank, S. (2008, July 6). Testing the Surprisal Theory of Word-reading Time, to be presented at *Annual Summer Interdisciplinary Conference (ASIC).* Abstract retrieved from http://www.cogs.indiana.edu/asic/2008/abstracts.asp

Friederici, A. D., Steinhauer, K., and Pfeifer, E. (2002). Brain signatures of artificial language processing: Evidence challenging the critical period hypothesis. *PNAS*, 99(1):529-534.

Gebauer, G. F. and Mackintosh, N. J. (2007). Psychometric intelligence dissociates implicit and explicit learning. *Journal of experimental psychology. Learning, memory, and cognition*, 33(1):34-54.

Grossman, E. R. F. W. (1953). Entropy and choice time: The effect of frequency unbalance on choice-response. *The Quarterly Journal of Experimental Psychology*, 5:2, 41-51.

Hale, J. (2001). A probabilistic Earley parser as a psycholinguistic model. *Proceedings of NAACL*, vol. 2: 159–166.

Hick, W. E. (1952). On the rate of gain of information. *The Quarterly Journal of Experimental Psychology,* 4, 11-26.

Hoen, M. and Dominey, P. F. (2000). ERP analysis of cognitive sequencing: a left anterior negativity related to structural transformation processing. *Neuroreport*, 11(14):3187-3191.

Jelinek, F. and Lafferty, J. D. (1991). Computation of the probability of initial substring generation by stochastic context-free grammars. *Computational Linguistics*, 17(3):315-323.

Jurafsky, D. and Martin, J. H. (2008). *Speech and Language Processing (Second Edition)*. Prentice Hall.

Klein, D. and Manning, C. D. (2005). Natural language grammar induction with a generative constituent-context model. *Pattern Recognition*, 38(9):1407-1419.

Levy, R. (2008). Expectation-based syntactic comprehension. *Cognition*, 106:1126-1177.

Lewicki et al. (1987). Unconscious acquisition of complex procedural knowledge. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 13(4):523-530.

Lewicki, P., Hill, T., and Bizot, E. (1988). Acquisition of procedural knowledge about a pattern of stimuli that cannot be articulated. *Cognitive psychology*, 20(1):24-37.

Lieberman, M. D., Chang, G. Y., Chiao, J., Bookheimer, S. Y., and Knowlton, B. J. (2004). An event-related fmri study of artificial grammar learning in a balanced chunk strength design. *Journal of cognitive neuroscience*, 16(3):427-438.

Nowak, M. A., Komarova, N. L., and Niyogi, P. (2002). Computational and evolutionary aspects of language. *Nature*, 417(6889): 611-617.

Opitz, B. and Friederici, A. D. (2004). Brain correlates of language learning: the neuronal dissociation of rule-based versus similarity-based learning. *Journal of Neuroscience*, 24(39):8436-8440.

Shanks, D. R., Johnstone, T., and Staggs, L. (1997). Abstraction processes in artificial grammar learning. *The Quarterly Journal of Experimental Psychology Section A*, 50(1):216-252.

Smith, N. and Levy, R. (2008). Optimal Processing Times in Reading: a Formal Model and Empirical Investigation. To appear in *Proceedings of the 30th Annual Meeting of the Cognitive Science Society* (oral presentation).

Stolcke, A. (1995). An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics, MIT Press for the Association for Computational Linguistics*, 21.