ORDER AND WORD ORDER

How the Information Content of a Word in a Sentence Helps Explain a Linguistic Universal

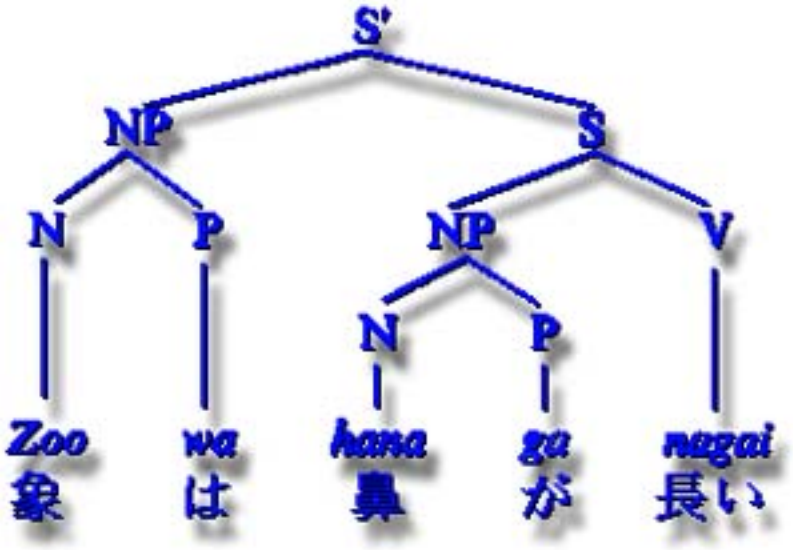


Image Courtesy Dept. of East Asian Languages and Literature, Ohio State University

In Fulfillment of the Thesis in Symbolic Systems

Gregory Wayne
Stanford University

2005

Acknowledgments

I am indebted to many people for this thesis.

Without Peter Lubell-Doughtie, I would never have known that there were people in the world researching the cultural and biological evolution of language.

Elizabeth Coppock, and Tony Tulathimutte were both extraordinarily helpful and supportive while I was switching between thesis topics to learn about language evolution during Symbolic Systems Honors College. Todd Davies generously helped me at all steps along the way.

I would like to thank Daniel Ford for discussing information theory with me when I knew I needed it but did not know exactly what it was. And I would like to thank Roger Levy for discussing information theory with me after I did and, especially, for turning me on to the entropy rate measure. Chris Manning, with whom I discussed the protocols of experimental design, was a great aid.

Logan Grosenick and Arvel Hernandez were both handy veterans when it came time to apply ANOVAs and retrieve significance measures.

My advisor, David Beaver, has been absolutely wonderful. David always seemed to manage to come to realizations that had taken me weeks within minutes. And among his smaller achievements, I have become a much better Internet searcher by watching him at work. I would also like to thank him for his personal kindness, perhaps shown best in setting up the language evolution reading group in conjunction with Hal Tilly. David's critiques of the papers we read were always dead-on, and I learned a great deal by listening. I wish him a full and speedy recovery from his recent illness.

Tom Wasow is a fantastic educator, and I am hard pressed to express my gratitude to him. His work in creating the Symbolic Systems Program is one of the few Absolute Goods to which a Symbolic Systems major might assent. In addition, his comments on my rough drafts have improved the writing and logic tremendously.

I would like to thank the close friends I have known over the last four years. There are many, but I would single out Kiel Downey, Rachael Norman, Alex Kehlenbeck, and Ross Perlin. My current roommate, Ryan Hebert, is an übermensch, a braggart warrior, a co-conspirator in deception, a wizard, a hipster, and a goof.

My older siblings–Teddy, Elizabeth, and Geoffrey–are very important to me, and I continue to model myself on them. My grandmother Bess is the sweetest person on Earth, and I would also do well to follow her life example.

My parents have financially obligated me to reserve the strongest warmth for them. They are still my intellectual and personal heroes. And I must give them credit for teaching me (in order) language and analysis. They have largely forgotten to teach me synthesis, but for that I forgive them.

Disacknowledgments

To the Python programming language: you make coding really easy, but you might want to speed up a little.

Introduction

Legend has it that the voracious linguist Joseph Greenberg would read the grammar of a different language each week. One can imagine that during one particularly absorptive session in 1963, Greenberg burst forth with the hypothesis that there are universal constraints on the grammars of all languages. By this time, it had been noted that the basic word types were good candidates to be considered linguistic universals. The universality of word types could be assessed in part by asking whether a language possessed a natural class of words which all together translated into words of the proposed class in English. If one wanted to know whether Nahuatl had nouns, one would go about answering the question by asking a native-speaking informant what classes of words there were in Nahuatl. Then one would try to determine if one of those classes overwhelmingly mapped into the English class of nouns.

However, Greenberg was after bigger game. Out of the seemingly disordered diversity in human languages, he noted some peculiar regularities. Few languages possess sentence-initial direct objects in their basic word order. In most languages, the subject of the sentence, the agent, instead comes first. Less frequently, in a minority of languages, the verb precedes the rest of the sentence. This distribution of word order is remarkable. Unfortunately, its cause can only be speculated upon. Greenberg himself remained agnostic about the sources of these word order distributions, but his explanatory lapse is forgivable: he had a series of even more brilliant insights immediately thereafter.

One of those insights is the focus of this thesis. The discovery was that *the ordering of the elements of Subject, Verb, and Object in the sentences of a language typically dictates the ordering of other word types within their respective phrases.* By knowing whether a language privileges its verb before its subject, for example, one can usually infer that the language utilizes *prepositions* as opposed to *postpositions*. Given the essential choice of word order ruling the subject, verb, and object, a cascade of consequences delimits the possible positions of relative clauses, genitives, adverbs, and so forth.

Depending on your frame of mind, this implicational principle is either a tool or a clue. As a tool, it permits one to codify a language's grammar neatly and expressively. As a clue, it gives

insight into how human beings actively and passively process utterances, learn how to speak languages, and evolved language in the first place. The first of these projects–that of using linguistic universals notationally–falls within the purview of linguistic typology; the second–that of using linguistic universals to measure theories of cognition–falls within language evolution. My own interests and work fit more squarely within language evolution, and I have looked at cognitive bases that might explain Greenberg's word order universals.

This thesis will be comprised roughly of three parts. The first will be an overview examining more modern typological characterizations of Greenberg's word order universals and surveying existing evidence for them in human languages. I will also look at scholarship within the field of language evolution that has attempted to explain the predominating word order patterns according to the criterion of learnability. The reasoning goes that the word order patterns that are most frequently attested in human languages are actually simpler to induce grammatically from linguistic data. I will then describe my own experimentation in this vein and conclude that past research on the learnability of a grammar has failed to explain the word order distributions of natural languages. And I will present a new argument that the word order correlations we find in natural languages are actually due to optimality considerations: grammars that prescribe frequently attested word orders generate sentences that communicate information at higher, more constant rates.

Overview

Greenberg's original paper (1963) outlined a series of about thirty universals, of which seven were related to word order. Later work refined these universals. Baker (2001) gives a pithy unification that he calls the *Head Directionality Parameter* (HDP): if a language emphasizes fixed word order, it tends to be either left-headed or right-headed. Put another way, languages tend to branch in a *consistent* manner. If one phrase type is left-headed, it is likely that the other phrase types will be, too.

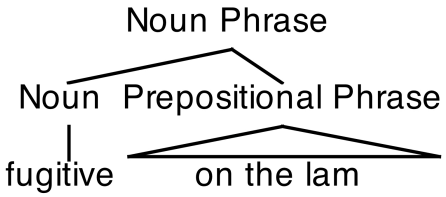


Figure 1

Figure 1 shows a left-headed phrase. The head of the noun phrase, the noun, is on the left, and the rest of the phrase branches recursively. It is important to understand how the HDP generalizes Greenberg's observations. One of Greenberg's universals states that in languages where the verb precedes the object, the language has prepositions as opposed to postpositions. The HDP would analogously conclude that, since the language has left-headed verb phrases, it will also probably have left-headed "adpositional" phrases (prepositions).

Now armed with a clearer understanding of the universal under scrutiny, what evidence is there for it? Greenberg's initial paper only examined 30 languages, drawn from European, African, Asian, Oceanian, and American Indian genera. In pooling these groups, Greenberg attempted to sample languages that were relatively independent and representative of the world's linguistic diversity; however, he confided that for convenience he also chose grammars from languages that were accessible or familiar to him. The Rosetta project, which aims to archive grammars and dictionaries for all the languages in the world, estimates there to be some 7,000 odd languages. 30 languages alone is not statistically meaningful. Moreover, it would be prudent to benchmark the hypothesis against another set of languages that Greenberg had not inspected while fishing for the conjecture.

Dryer's Investigation

Dryer (1992) conducted a more sedulous survey of 625 languages for which data were available, looking for word orders that correlate with the ordering of verb and object. More precisely, he gave the following definition for a correlation:

> I will refer to the ordered pair <X, Y> as a **correlation pair**, and I will call X a **verb patterner** and Y an **object**

> **patterner** with respect to this correlation pair. For example, since OV languages tend to be postpositional and VO languages prepositional, we can say that the ordered pair <adposition, NP> is a correlation pair, and that, with respect to this pair, adpositions are verb patterners and the NPs that they combine with are object patterners. (Dryer, p.82)

The 625 languages were grouped into 196 genera. Genera were taken to be groups of languages whose relatedness is uncontroversial. Dryer first inspected these 196 genera for pre- or postpositions. If the majority of languages within a genus had prepositions, then the entire genus was considered to be prepositional. The results were as follows:

	OV	VO
POSTP	107	12
PREP	7	70

Figure 2

This is a promising result. It shows that the placement of the adposition (the existence of prepositions or postpositions) does correlate strongly with the ordering of verb and object. Unfortunately, there are confounding factors. Languages within the same regions are likely to have similar properties for a variety of reasons, among them cultural transmission and common lineage. The observed correlation may not be for any deep reason at all. Dryer took pains further to class the genera based on their region of genesis (Africa, Eurasia, Southeast Asia and Oceania, Australia and New Guinea, North America, South America). As with grouping in genera, if the majority of genera within a class had property A, then the entire class should be marked with property A. Presumably, if the results were still significant within these new groupings, it would not be due to geographical or historical factors.

Dryer looked at 21 candidate correlation pairs and found that 16 showed significant correlation with the ordering of verb and object. Notably, the 5 non-correlating types were less crucial features of language; for instance, the relative positioning of intensifier and adjective (e.g., "*very big*") does not correlate with the relative positioning of verb and object. Importantly, most of the essential classes of correlation pairs that Greenberg had pointed to did correlate significantly with the sequencing of verb and object.

Of course, these must be remembered as *tendencies*. Some languages like Warlpiri or Nuggubuyu do not exhibit hierarchical constituent structure at all but embed syntactic relationships in morphology (Dryer, 1992). Furthermore, the word order of most languages is somewhat flexible. English, a paradigmatic left-headed SVO language, also licenses VS constructions–most commonly in stage directions! Some languages even tolerate use of all 6 possible orderings of subject, verb, and object in sentences (Van Everbroeck, 2003). The grammatical codifications from which Dryer drew his data must have universally simplified their characterizations of the languages' word orderings. Still, with these caveats in mind, all evidence points to the existence of real correlations among word orderings in the phrases of syntactical languages.

Dryer advocates strongly against descriptions of these correlations which invoke head-dependency. The reader may recall that the HDP posits that languages tend to order words within phrases according to whether the "head" is on the left or the right. Dryer argues that head-dependent theories predict languages will exhibit fallacious word order correlations. For example, English is left-headed but has the noun phrase rewrite rule NP → Det N PP, where Det denotes a determiner. In this case, the noun is actually not on the left side of the phrase–the determiner is–even though the noun is the phrase's head. Dryer hopes to fix this problem by abstracting away from linguistic heads. His *Branching Direction Theory* (BDT) demands fewer assumptions and fits the data better. It says:

> A pair of elements X and Y will employ the order XY significantly more often among VO languages than among OV languages if and only if X is a nonphrasal category and Y is a phrasal category. (Dryer, p.114)

Framed this way, the BDT permits an English determiner to precede the noun in the noun phrase because it is a nonphrasal category. It is sufficient that both the determiner and the noun be to the left of the prepositional phrase.

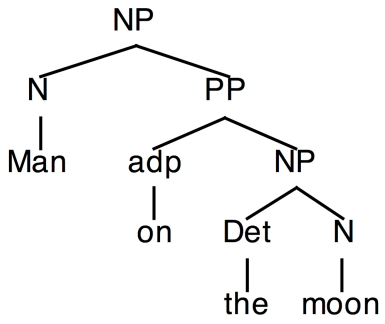


Figure 3: The NP comprising "the moon" is tolerated by the BDT. Neither "the" nor "moon" is a phrasal category. The BDT therefore does not necessitate that "moon" precede "the." A head-dependent theory would argue that "moon" should precede "the" in a left-headed language.

It is beyond the scope of this paper to decide among different descriptive formulations of the word order correlations. The BDT, however, is a more formal theory, differentiating between left and right-ordered constituents based purely on a structural consideration (is the constituent a recursively branching phrase?). For this reason, the BDT can be more conveniently tested on computers, and this paper will endeavor to explain the Greenbergian word order correlations within the BDT framework. In cases where the head of a phrase is unambiguous and there is no disagreement with the BDT, I will sometimes continue to refer to left-branching phrases as right-headed and right-branching phrases as left-headed.

Although Dryer labels the BDT a theory, it should more appropriately be called a predictive description. In itself, it concisely explains the data. However, it provides no causal intuition as to why the world's languages demonstrate these word order patterns. Ideally, we would like a theory to ground the observed phenomena in fundamental features of language or cognition.

In general, before issuing any hypotheses, it is good to describe how a linguistic universal can arise, or, analogously, how two languages can happen to share similarities. The following list is meant to be as exhaustive as possible:

I. Value: The space of values for a property is so small as to confine most languages to the same value. *Prima facie* branching direction could take on a vast number of values since every phrase in a language could branch in a direction independent of the others. The fact that languages are mostly left-branching or right-branching is not therefore explained by lack of opportunity; and, given the enormous range of possible branching patterns over all phrases in a grammar, it is unlikely that languages cluster towards the poles of left and right branching consistency as a matter of chance.

II. Shared genealogy: Two languages are similar because they are related. Both languages inherited a trait from a progenitor language. Van Everbroeck (2003) remarks that the distribution of language types today may derive from founder effects from the typological distribution of languages thousands of years ago. Depending on how quickly a language can change its branching direction, this idea cannot be ruled out. If branching direction tends to maintain its features for millennia, the current distributions of branching direction may well be artifacts of the distant past. This said, I am not aware of any research characterizing rates of change in branching direction.

III. Borrowing and cultural transmission: Communities in areas bordering a prestige or dominant language can adopt the traits of the dominant language. Dryer addresses this possible confound by grouping languages based on geography. Nevertheless, neither cultural transmission nor shared genealogy can explain why languages tend towards branching consistency. A mixed branching language could just as easily transfer its mixed branching traits to surrounding language communities and successively evolving versions of itself.

IV. Shared environmental constraints: It is difficult to imagine that the physical environment within which a language is spoken could have an observable effect on the language's syntax. However, it easier to imagine that the tasks that a language is used for might privilege different types of syntax. For example, in domains where the subject is more important to meaning than the verb, a subject-first word ordering might be employed. In cooperative hunting, for example, it might be important to indicate the subject first before the verbal action, in the same way that ballplayers will shout a

teammate's name, leaving the action of passing the ball implied. These are somewhat far-out ideas, and I mention them merely for the sake of completeness.

V. Shared cognitive constraints: All languages are spoken by humans. Memory, processing (Hawkins, 1990 and Kirby, 1998), and learning limitations (Christiansen and Devlin, 1997) have all been raised as possible explanations for the existence of branching consistency. If consistent grammars are easier to learn, or consistent sentences are easier to comprehend and generate, then human populations may favor grammatical simplicity.

VI. Semiotic constraints: Deacon (2003) argues that many linguistic universals may arise from constraints on any sufficiently complex symbolic system used for human communication. In the same way that prime numbers in mathematics were neither discovered nor constructed (but followed logically from the definition of integer division), various features of language may neither be discovered nor constructed, necessarily emerging from the constraints and easements of any functional system for communication. Deacon gives "non-degrading recursivity," the fact that natural language grammars are more or less context-free, as an example of a feature of language that is demanded of any symbolic system capable of non-trivial representation.

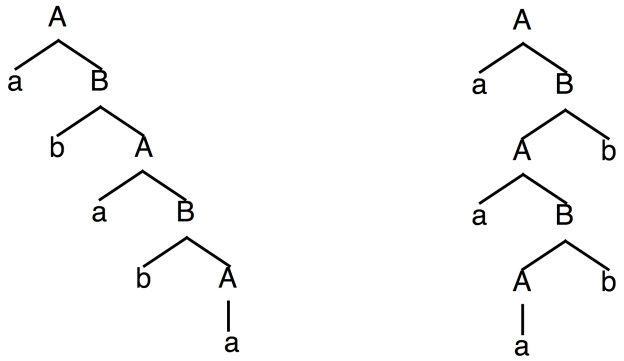
Some of these possibilities are beyond the grasp of science to answer. For example, if languages were distributed as left and right branching by chance, no amount of deliberation or experimentation could resolve this fact. However, if the predominance of branching consistency is caused by cognitive or semiotic constraints, then one should expect the answer to be more tractable.

Christiansen and Devlin (1997) argue that cognitive constraints have created the cross-linguistic tendency towards branching consistency. They theorize that natural languages proscribe against branching inconsistency because inconsistent grammars are more difficult to parse and learn. To illustrate this, they give the example of two mutually referential grammar rules. In one case, the rules branch consistently with each other, and in another case they are inconsistent. Because this example is central to understanding their thesis, I will reproduce it here.

G1: Consistent	G2: Inconsistent
A → a (B)	A → a (B)
B → b A	B → A b

Figure 4: the parentheses denote that the enclosed constituent is optional

These rewrite rules define one consistent and one inconsistent grammar. Grammar G1 generates sentences that are series of the alternating lower case letters 'a' then 'b'. Grammar G2 licenses sentences that are comprised of N 'a'-s, followed by (N - 1) 'b'-s.



Figures 5 and 6: Tree structures over sentences generated by G1 and G2, respectively.

The sentence generated by the inconsistent grammar, G2, has characteristic "center embeddings" that are created by the fact that the recursion traverses first the right constituent then the left constituent as it expands phrases. Christiansen and Devlin state that "center embeddings are difficult to process because constituents cannot be completed immediately, forcing the language processor to keep material in memory," (p.2). It is not at all immediately obvious what this means from the article; however, the theory derives from prior work by Hawkins (1990). Essentially, Hawkins's theory says that the human language processor wants to parse the constituents of phrases that uniquely determine the phrase type first, allowing important syntactic relations to be fixed as soon as possible. In an English noun phrase, the first constituent that uniquely identifies the phrase type is the determiner; in a verb phrase, the auxiliary or main verb identifies the phrase type uniquely. In the case of Figure 6, the head of the first phrase B is not parsed until the end of the

sentence, creating ambiguity as to what kind of phrase follows the sentence-initial 'a'. (I will later describe another theory that is in many ways more parsimonious. For the moment, the explanation given by Hawkins should suffice.) Christiansen and Devlin further posit that the difficulty in processing center-embedded sentences should make acquisition of center-embedded constructions more difficult because syntactic dependencies may span longer distances within sentences, making the associations between syntactic elements less obvious to the learner.

Christiansen and Devlin's Simulation

To test this assumption, they decided to define grammars with different degrees of branching consistency to see if an artificial learner–as a model of a human learning natural languages–would have more difficulty learning the inconsistently branching grammars. Computational simulation of the grammatical learning process isolates learning from all other phenomena that might influence the typological tendency towards branching consistency.

Christiansen and Devlin first defined a "grammar skeleton." A grammar skeleton represents a number of different grammars at the same time. Figure 7 shows their chosen grammar skeleton. The brackets denote that the first constituent in the bracketed phrase can be placed either head-first or head-last. Thus, VP → V (NP) (PP), or VP → (NP) (PP) V.

Skeleton
S →NP VP
NP → { N (PP) }
PP → { adp NP }
VP → { V (NP) (PP) }
NP → { N PossP }
PossP → { Poss NP }

Figure 7

If all of the phrases appear with the ordering shown in the diagram of the grammar skeleton, the grammar generated is right-branching. Since there are 5 phrases that can be flipped,

there are $2^5 = 32$ different grammars total. They also defined a measure of inconsistency, counting the number of mutually referential phrases that are inconsistent. For example, the phrase combination NP → PossP N and PossP → Poss NP would increase the inconsistency measure of the grammar by 1. They labeled each grammar with an inconsistency measure and randomly generated a large number of sentences using each grammar. Although these grammars are very artificial simplifications of natural language grammars, they are complex enough to encode interesting rule interactions that modify branching consistency.

A simple recurrent neural network (SRN)—or connectionist model—was used to learn each grammar. An SRN is a generalization of the standard "feedforward" neural network (Elman, 1990), which was first put forth as a very reduced model of networks of neurons in the brain.

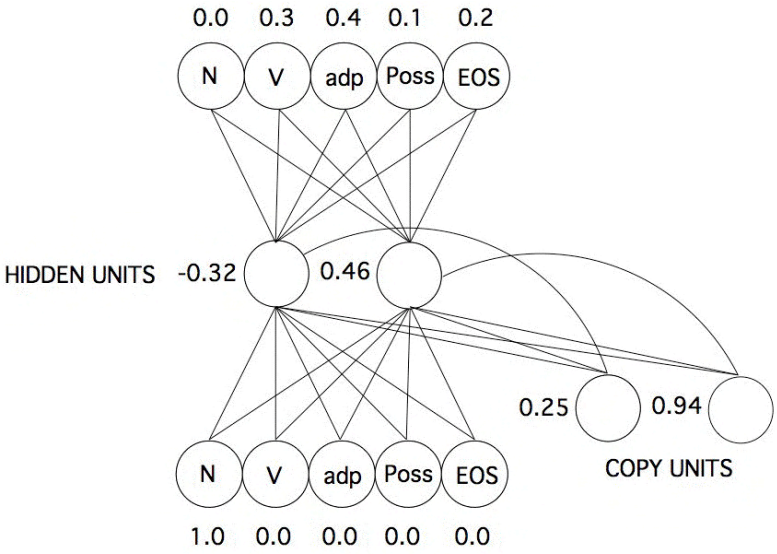


Figure 8: An SRN with a noun as input. Each input node represents a word token. The outputs represent a probability distribution over the next word in the sentence. The copy nodes hold the activations of the hidden nodes from the previous time step. The arcs between the hidden and copy nodes represents the back-copying of activations from the hidden layer to the copy layer at the end of each time step. 0.25 and 0.94 indicate the activations of the copy nodes at the current time step and the activations of the hidden nodes at the previous time step. EOS stands for End-of-Sentence marker.

Figure 8 diagrams a simple recurrent network like the one used in Christiansen and Devlin's simulation. The network works iteratively in discrete time steps. At each time step, the

next word in the sentence–a noun in the diagram–is input as vector, e.g., {1.0, 0.0, 0.0, 0.0, 0.0} for noun as shown in Figure 8, to the input nodes. Each of the lines between nodes represents a real-valued weight, which is multiplied against the value of the node it is connected to. The updated value or activation of a given non-input node i is computed as $g\left(\sum_{\forall j \in A_i} w_{ij} y_j\right)$. Here, A_i is the set of all nodes anterior to the node i, w_{ij} is the weight of the connection between node j and node i, and $g: R \rightarrow \{x: 0 \leq x \leq 1\}$ normalizes the net input to a node to between 0 and 1. The copy nodes contain the activation of the hidden nodes from the previous time step. They are similar to input nodes in that their output feeds into the hidden nodes. At the end of the time step, two actions occur. The actual next word is compared against the distribution of predicted next words, and the network is trained to increase the probability of the seen word type, given the words it has seen already in the sentence. This training is done by changing the network connection weights via the backpropagation learning algorithm. Then, as designated by the arcs in Figure 8, the activation from the hidden nodes is transfered to the copy nodes, thereby preserving the old activations for the next time step. This gives the network a limited "memory," allowing the activations of the network at the previous time step to feed into the computation of output for the current time step. A standard feedforward neural network has no copy nodes and therefore no memory. The memory in the copy nodes allows the SRN to make different predictions of what the next word will be, predicated on the words it has already seen in the sentence so far.

Large corpora of sentences were randomly generated from each grammar, and the SRNs were trained on the sentences. To determine if the networks had learned the grammars correctly, the output distributions of the neural networks given the sentential context were compared against the empirical conditional probability distributions in the training corpus using the mean-squared error metric (MSE), calculated as $\sum_i (p_i - m_i)^2$ for discrete probability distributions. p_i indexes the empirical distribution (the corpus's), and m_i indexes the model distribution (the neural network's output distribution). According to Manning and Schütze (1999), this is actually the wrong metric to use for a somewhat technical

reason[1]. Instead, the preferred metric is the so-called cross-entropy metric, which is the standard measure of deviation between two probability distributions. Whether their results remain credible, I cannot be sure, given that the original data are not presented in the paper. In any event, they reported a strong and consistent positive correlation between branching inconsistency and an SRN's deviations from the empirical probability distribution. They concluded that the SRNs had more difficulty learning inconsistent grammars, thereby bolstering the claim that inconsistent grammars are rare because they are more difficult to learn.

In addition to the analytical error of substituting MSE for cross-entropy, Christiansen and Devlin committed a far greater sin. Implied in their simulation was the assumption that "SRNs constitute viable models of [human] natural language processing" (p.5). However, Christiansen and Devlin were unaware or neglectful of the fact that SRNs perform particularly poorly at sequential prediction tasks. SRNs do not stand up as sufficient models for human grammatical learning because there are other recurrent neural network architectures that outperform them by a large margin. If connectionist modeling results are to be believed, then at the very least the most accurate models should be used.

The SRN Versus LSTM

Every time the network predicts the next word in a sequence, it is presented with a target vector, which encodes the actual next word in the sequence. The difference between the predicted probability distribution and the target vector is the net's error (calculated as the

---

[1] Say the network reports a probability of m(current word | previous words) = 0.0 for a word token, and the empirical conditional probability distribution is actually p(current word | previous words) = 0.1. Then the network reports a likelihood of 0.0 for the entire sequence previous words + current word (calculated as the chained multiplication of the conditional probabilities for each word in the sequence). However, the actual likelihood is much higher. Thus, we should penalize the difference between m = 0.0 and p = 0.1 much more than the difference between m = 0.2 and p = 0.1, even though the network's prediction is off by the same arithmetic amount in each case. The MSE, calculated as the sum of squared differences, is not sensitive to this, whereas the cross-entropy is. In general, the cross-entropy more rigidly penalizes models that deviate in assignment on high and low probability events.

MSE). But a question arises: how much did the activation of each node in the network contribute to the faulty prediction? This is important information which we use to increase the strength of node weights that would have led to correct prediction and decrease the strength of node weights that were implicated in the error. In a standard feedforward network, which has no memory of the previous words in the sentence, the backpropagation algorithm solves this problem. In the case of a simple recurrent network, the activations from previous time steps are still affecting the predictions at the current time step (remember the copy nodes). In order to make correct predictions, another more nettlesome problem must be solved: how much did the activation of a node at an arbitrary previous time step affect the error at the current time step?

It turns out that the SRN is not well equipped to handle the task of computing the errors due to nodes at prior time steps. In fact, according to Hochreiter and Schmidhuber (1997), error signals from distant time steps vanish. Effectively, the SRN abides by a statute of limitations; to the SRN, it is as if errors made many time steps ago never occurred at all yet their effects persist–the SRN can never learn to overcome mistakes that were introduced many steps before the final error. This is particularly problematic when it comes to learning grammars. The task of learning a grammar amounts to inducing hierarchical, long-distance dependencies between words in the sentential sequence. Figure 9 shows an example of a syntactic dependency that spans multiple words.

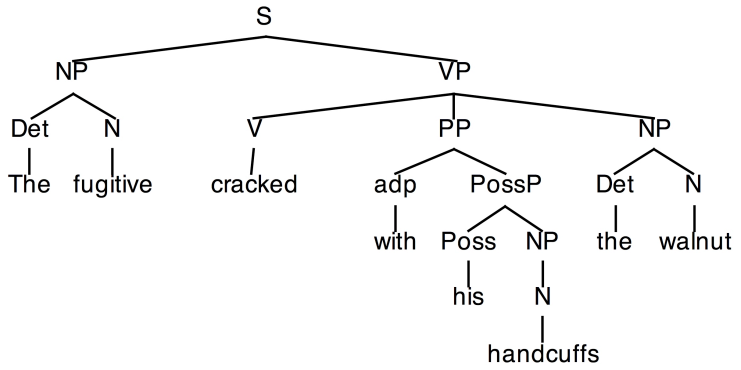


Figure 9: To predict that an object NP is coming after the PP "with his handcuffs," a recurrent neural network must reliably retain the information that it has already seen the transitive verb "cracked" four words before. The SRN has difficulty adapting its behavior to make predictions based on relatively distant context.

Hochreiter and Schmidhuber designed a novel recurrent neural network architecture that they called "Long Short-Term Memory" (Hochreiter and Schmidhuber, 1997). Unlike the SRN, it circumvents the problem of vanishing error flow and is able to learn to consider more distant contextual information in its predictions. On the standard task of learning the Embedded Reber Grammar, the SRNs were unable to learn the grammar in any trial, while the LSTM networks were successful 100% of the time. In 5 other tasks involving long sequence classifications, LSTM clearly outperformed any other recurrent network architecture, including the SRN. Additionally, LSTM has a biological plausibility that the SRN does not (Graves, Eck, Beringer, Schmidhuber, 2004). The SRN algorithm must explicitly copy the activations of the hidden nodes into the copy nodes at each time step—an action that is difficult to imagine real neuronal populations performing because it involves a global copy of activity from neuron to neuron. LSTM, on the other hand, does not copy activations between nodes. Moreover, the LSTM algorithm has computational complexity that is "local in space." That is, the number of operations to compute one time step scales linearly with the number of neurons in the network.

Experiment 1: LSTM and Simulated Grammar Learning

Although branching inconsistency compromised grammatical learning for SRNs, it was unclear that branching inconsistency would have any effect on LSTM grammar learning. An experiment to determine the effect of grammatical inconsistency on LSTM learning was performed. 32 grammars were created according to the grammar skeleton in Figure 7. Each grammar generated 10,000 sentences with uniform probabilities associated with each of the rewrite selections. 32 LSTM networks were instantiated, one for each grammar. The topology of each LSTM network was as follows: 5 input units (one for each terminal category, including the end-of-sentence category), 5 output units (also one for each terminal category), 0 hidden units, 3 blocks with 2 cell block size, 0.1 learning rate, and all weights initialized randomly with absolute value less than 0.1. Each network was trained on one of the 32 grammars. Training amounted to seeing the corpus of 10,000 sentences two times to prevent overfitting.

Each network was then tested on a third run through the corpus. At each sentence symbol, the cross-entropy was taken between the network's predicted distribution for the next symbol and the corpus's conditional probability distribution over the next sentence symbol given the words seen so far in the sentence. Since some grammars might have generated more words total than others through longer sentence productions, the cross-entropies at each word were totaled and divided by the number of words in all the sentences of the grammar's corpus. This per word cross-entropy statistic was used to measure how well a network had learned a grammar–with low scores indicating that the network's word predictions closely approximated the conditional probability distributions in the corpus.

The measure of a grammar's inconsistency was taken from Christiansen and Devlin (1997). They defined the Recursive Rule Interaction Constraint score (RRIC). If two mutually referential rules are inconsistent, then the RRIC is incremented by 1. This gives a maximum RRIC of 2. Additionally, because a "PP can occur inside both NPs and VPs, a RRIC violation within this rule set is predicted to impair learning more than a RRIC violation within the PossP recursive rule set" (p.4). They therefore incremented the RRIC by another 1 in the case of recursive inconsistency in the PP rule set. This seemed a

relatively unmotivated amendment to the RRIC, and I included it only to make my results comparable to theirs. Given the aforementioned amendment, the maximum RRIC is 3.

Results showed no significant effect introduced by inconsistency. The correlation between RRIC and per word cross-entropy was -0.334, which, if anything, would show that inconsistency increased learnability (decreasing the LSTM network's deviations from the empirical conditional probability distributions). A one-way ANOVA demonstrated the significance level to be 0.282 with $F(3, 28) = 1.340$, indicating that there is a 28.2% likelihood that the RRIC classes {0, 1, 2, 3} would show greater inter-class variance purely on the basis of chance (by the null hypothesis). Levene's test for homogeneity yielded a significance of 0.895, which indicates that the ANOVA assumption of homogeneity between groups is met. A power analysis revealed that sample size was sufficient to find differences between groups if they existed to a power of 0.809.

Taken together, these results show that the LSTM network's grammatical learning performance is unaffected by the recursive inconsistency of the grammar within the parameters of this experiment. Simulations from realistic grammar skeletons with actual word terminals would be useful. Unfortunately, current connectionist architectures, including LSTM, are not up to the task of learning anything but the simplest probabilistic context-free grammars (Gers and Schmidhuber, 2001); when word terminals are added to the grammar, the number of input nodes climbs, raising the number of parameters the network needs to fit, and dramatically increasing the necessary number of example sentences—the so-called "curse of dimensionality."

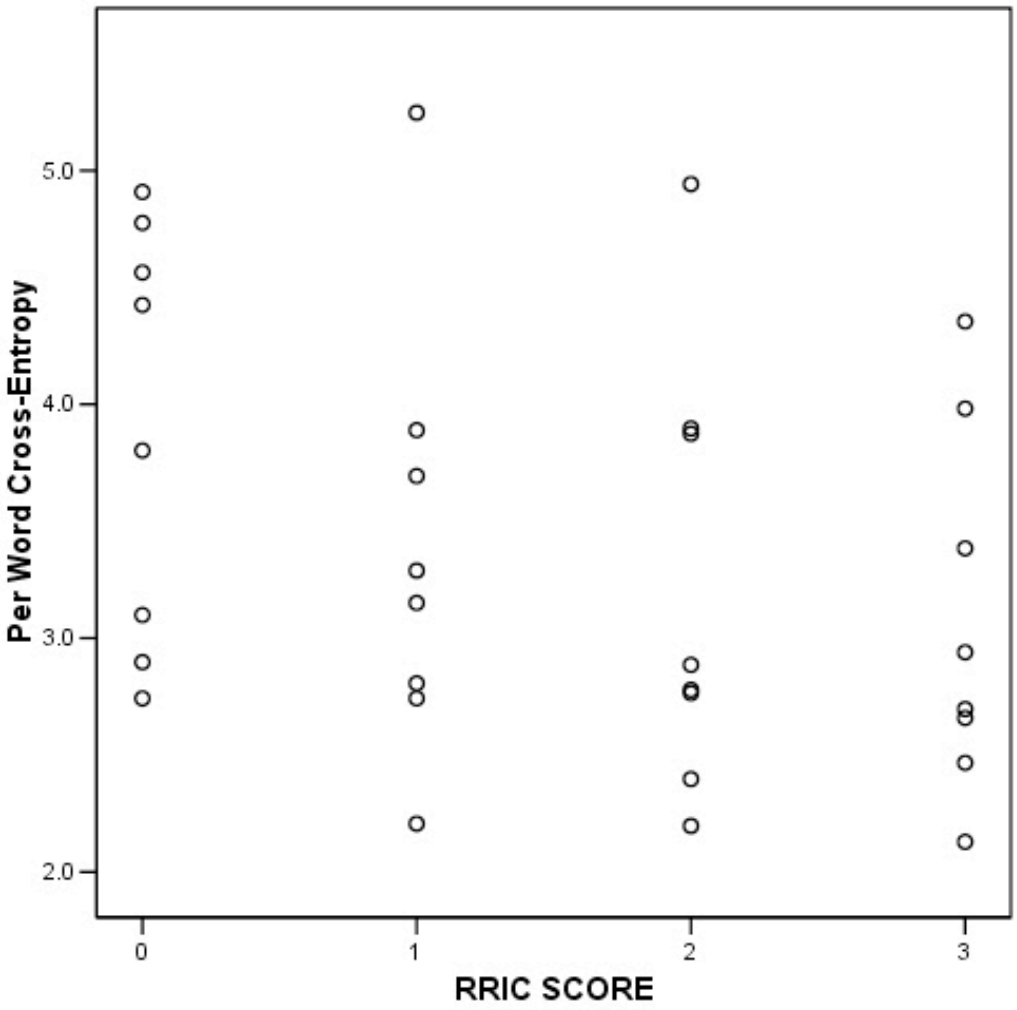


Figure 10: The branching inconsistency of a simulated grammar is uncorrelated with the difficulty of learning that grammar.

And yet other convergent research lends plausibility to the claim that inconsistent grammars are hard to learn. Christiansen has studied "artificial language learning" tasks in which one group of human subjects is given positive examples of an inconsistent language and another group is given samples from a consistent language as training data (Christiansen, 2002). After training, subjects were told to make grammaticality judgments on a third set of unseen examples. Subjects who had seen the consistent sentences performed better, supporting the theory that consistent grammars are easier to learn. However, it is difficult to evaluate how closely artificial language learning tasks replicate human language learning: among a great many possible criticisms, only 30 sentences were

used as training data; only adults past their critical periods served as experimental subjects; and the languages, whose sentences were strings of alphabetical characters, were learned by rote memorization of text, not by exposure to spoken sentences with associated meanings.

Eager Processing and Uncertainty

The evidence that consistently branching grammars are easier to induce is questionable. However, even if recursive inconsistency does impede learning, the question remains why. Christiansen and Devlin's work addresses the question of how grammatical consistency gets embedded typologically (possibly via learning), but it largely overlooks the specifics of how branching consistency interacts with human cognition. As mentioned before, Hawkins has made some headway in unpacking why consistent grammars are easier for human beings to parse (Hawkins, 1990). However, he builds his explanation around a formalism that makes awkward assumptions. One article of faith that Hawkins expounds is that "mother nodes [phrasal categories] will be constructed only when their presence is uniquely determined by the input" (p.227). Although this is possible, human cognition is certainly not obligated to perform the operation of processing language in the way that Hawkins imagines. Additionally, most of the parsing machinery that Hawkins envisions is non-probabilistic. Hawkins assumes that humans construct all possible parses of sentences at once; however, I think it is quite clear that in the case of syntactic ambiguity, human beings will sometimes conceive an incorrect parse to the exclusion of all others, unaware that the sentence could be construed another way. Hawkins's theory of human parsing therefore does not commit to the idea that human beings anticipate possible subsequent words as they comprehend sentences. This conflicts with many reasonable, more recent theories of intelligence like that one of another Hawkins–Jeff Hawkins. Jeff Hawkins argues that the brain is constantly making predictions about its future inputs, including linguistic ones (Hawkins and Blakeslee, 2004). Although I can hardly speculate on the conscious experiences of others, this is certainly closer to my own introspective experience.

Hale (2003) makes a more minimal set of assumptions about human sentence processing. One relevant assumption is that comprehension is "eager," meaning that "no processing is deferred beyond the first point at which it could happen," (p.105). Whereas Jack Hawkins's

parsing theory states that mother nodes aren't constructed until the point in the sentence at which they are uniquely specified, Hale's theory states that the human expends effort disconfirming impossible syntactic derivations. Therefore, humans continue to resolve ambiguity even in the face of uncertainty. Hale remarks that a probabilistic phrase structure grammar assigns probability 1.0 over all its derivations (by the definition of a probability distribution). The so-called prefix probability of a subsequence is the summed probability of all derivations that are consistent with that subsequence. The amount of work that a parser has done at a given point in a sentence is related to 1.0 minus the prefix probability (Hale, 2001). (I say "related" because the parser's work is more precisely specified by the reduction in *uncertainty* over derivations, a quantity that I will define in the next section.) Say a grammar[2] generates three sentences with equal probability: {"I want to be a rider like my father", "I want food", "Dunno"}[3]. The prefix probability of "I want" is 2/3. Therefore, the amount of work the parser has performed at the point of seeing "I want" is related to 1.0 - 2/3 = 1/3. The next word after "I want" will further reduce the derivational probability by 1/3 because the two sentences "I want to be a rider like my father" and "I want food" here diverge in form.

Now, if the amount of work that a parser must perform is the reduction in uncertainty over derivations that correspond to the sentence, one might expect that–all other things equal–grammars would evolve (culturally or genetically) to be as predictable as possible to lessen the effort by the parser (or the human beings who must utter and comprehend sentences from those grammars). The typological tendency towards branching consistency may arise because consistently branching grammars generate sentences that are more word-by-word predictable. If this is the case, then Greenberg's word order correlations can be explained in a very natural way.

---

[2] In this case, we are more correctly describing a language model than a grammar because we are assigning probabilities not to phrasal productions but to the sentences themselves.

[3] Besides the sentence "I want food," which is invented for the example, these are translations of what Kaspar Hauser, a famous feral child, was able to say when first questioned by authorities.

Entropy as a Measure of Uncertainty

Admittedly, all this discussion of uncertainty and predictability has been rather vague. It would be appropriate to make explicit some terms that formally define our measures of uncertainty before going forward. First, we define the surprisal S(x) of an event x:

$$S(x) = -\log p(x) \quad \text{(Eq. 1)}$$

This makes intuitive sense: events of probability 1 have surprisal 0; events of probability close to 0, have very high surprisal. (One is unsurprised by the occurrence of events that are likely, and one is very surprised by unlikely events.) The uncertainty over all the events in a probability distribution is the average surprisal over all events. This is known variously as the entropy, the uncertainty, or the information content of a probability distribution and is represented as H(X):

$$H(X) = \sum_{x \in X} -p \log p(x) \quad \text{(Eq. 2)}$$

This measure is maximized if all the events in the distribution have equal surprisal. This can be simply understood. If the event in question is the flip of a coin, and the coin is weighted such that the probability of heads = 0.95, and the probability of tails = 0.05, then uncertainty as to the result of the coin flip is small. Uncertainty about the result of a coin flip is maximized (high entropy) when heads and tails have equal probability of occurring. The uncertainty associated with the flip of a fair coin can be represented as H(0.5, 0.5) = 1 bit. This is the amount of information that is needed to specify uniquely the outcome of one flip of a fair coin. The uncertainty associated with the nth word X_n following a string of words can be represented by means of conditional entropy, which reads "the entropy associated with the nth word given the preceding words":

$$H(X_n \mid X_{n-1}, X_{n-2}, \ldots, X_1) \quad \text{(Eq. 3)}$$

A useful property of conditional entropies is that they can be "chained" arithmetically. The uncertainty over all symbols in a sentence is the uncertainty over the first symbol, plus the uncertainty over the second symbol given the first, plus the uncertainty over the third given the first two, and so on. This can be written as follows:

Entropy Chain Rule

$$H(X_n, X_{n-1}, \ldots, X_1) = H(X_1) + H(X_2 \mid X_1) + \ldots + H(X_n \mid X_{n-1}, \ldots, X_1) \quad \text{(Eq. 4)}$$

One measure closely related to conditional entropy is the entropy rate. For a stationary process (informally, one whose distribution over events does not change over time), this is defined as:

$$\lim_{n \to \infty} H(X_n \mid X_{n-1}, X_{n-2}, \ldots X_1) = \lim_{n \to \infty} \frac{1}{n} H(X_n, X_{n-1}, X_{n-2}, \ldots X_1) \quad \text{(Eq. 5)}$$

Utterances are not completely stationary processes. The probability of a preposition at the end of a sentence is very low in some dialects of English; in general, the probability of any given word varies throughout the course of the sentence. The changing probabilities of verbal events bear on conditional entropy: for example, the uncertainty of the last word of an utterance in that fictitious variant of Canadian English whose utterances always terminate in "eh" is 0.

Genzel and Charniak (2002) state that it "is well-known from Information Theory that the most efficient way to transmit information through noisy channels is at a constant rate," (p. 1). Without plunging deeply into discussion, if we assume that the speaker and listener can each process only a finite amount of information at a time, then one maximizes information throughput by passing information at a constant rate. From a parsing perspective, this amounts to saying that the work of the parser–disconfirming inappropriate derivations–is also spread out over the course of the sentence. It could be a maxim of conversation in the manner of Grice: to communicate as much information as possible, find the highest rate of information transfer at which it is possible to communicate reliably, and stick with that mark.

With these definitions in hand, it is now possible to explain branching consistency suitably. Consider again the two grammars in Figures 5 and 6. I mentioned before that G1 licenses sentences that are comprised of an alternating series of 'a' and 'b'. In particular, a sentence S1 of length 2N (including the end-of-sentence marker) consists of N 'a'-s alternating with N-1 'b'-s followed by the end-of-sentence marker (EOS). A sentence generated by G2 (S2)

consists of N 'a'-s in sequence, followed by N-1 'b'-s, then the end-of-sentence marker. Now, what is the entropy rate of G1? Although the entropy rate is formally supposed to be taken in the infinite limit, we will approximate it with finite length (again, utterances are never infinitely long).

$$\frac{1}{2n}H(X_{2n},X_{2n-1},...,X_1) = \frac{1}{2n}\left(H(X_1) + H(X_2 \mid X_1) + ... + H(X_{2n} \mid X_{2n-1},...,X_1)\right)$$

By the Entropy Chain Rule

$$= \frac{1}{2n}\left(H(X_2 \mid X_1) + H(X_4 \mid X_3,X_2,X_1) + ... + H(X_{2n} \mid X_{2n-1},...,X_1)\right)$$

Because the sentence can only end on an 'a', another 'a' is predictable whenever the parser is on 'b'

$$= \frac{1}{2n}\left(H(\tfrac{1}{2},\tfrac{1}{2}) + ... + H(\tfrac{1}{2},\tfrac{1}{2})\right)$$

Either the sentence terminates, or there is another 'b'

$$= \frac{n}{2n}H(\tfrac{1}{2},\tfrac{1}{2})$$

There are n predictions, whose choices are between 'b' and EOS

$$= \frac{n}{2n} = 1/2$$

By the definition of Entropy

For S1, the only uncertainty is whether a 'b' or EOS will succeed an 'a'. If the parser has seen a 'b', there is no uncertainty about whether an 'a' will follow: it will. Therefore, the conditional entropy alternates between 1 and 0 with every additional symbol. S2 has a different profile.

$$\frac{1}{2n}H(X_{2n},X_{2n-1},\ldots,X_1) = \frac{1}{2n}\left(H(X_1) + H(X_2 \mid X_1) + \ldots + H(X_{2n} \mid X_{2n-1},\ldots,X_1)\right)$$

By the Entropy Chain Rule

$$= \frac{1}{2n}\left(H(X_1) + H(X_2 \mid X_1) + H(X_3 \mid X_2,X_1) + \ldots + H(X_{n+1} \mid X_n,\ldots,X_1)\right)$$

After seeing 1 'b', one automatically knows that n - 2 'b'-s remain;
The only uncertainty lies in whether an 'a' will be followed
by a 'b' or another 'a'

$$= \frac{1}{2n}\left(H(\tfrac{1}{2},\tfrac{1}{2}) + \ldots + H(\tfrac{1}{2},\tfrac{1}{2})\right)$$

$$= \frac{n}{2n}H(\tfrac{1}{2},\tfrac{1}{2})$$

There are n times when it is uncertain whether the next symbol
will be 'a' or 'b'

$$= \frac{n}{2n} = 1/2$$

By the definition of Entropy

Once the parser has seen the first 'b', it knows that only 'b' can follow. And it also knows that there will be N-1 'b'-s total. Overall, one-half bit of information is conveyed by each symbol in both S1 and S2. Despite the fact that the entropy rate is the same for S1 as it is S2, the entropy rate is constant over the course of S1; however, the entropy rate for S2 begins very high (with every symbol uncertain) and drops to 0 once the first 'b' has been seen.

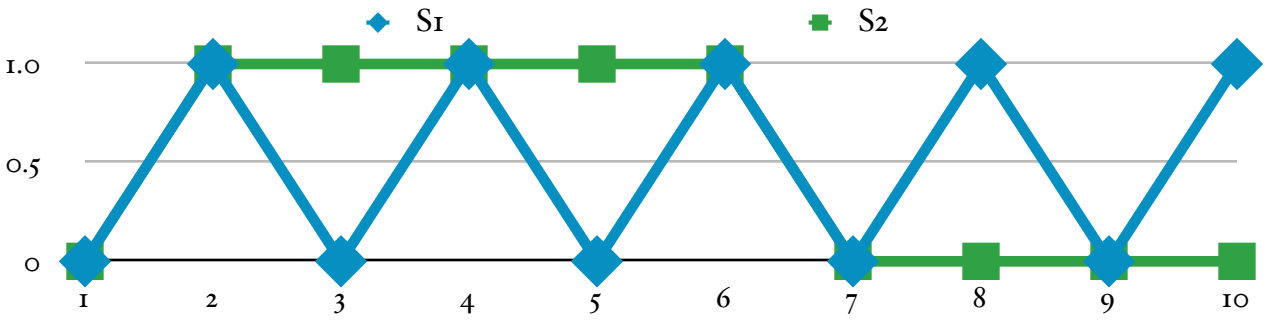


Figure 11: The entropy rates (in bits) of two sentences, each of length 10, generated by G1 and G2. The entropy rates for sentences S1 and S2 are the same over the course of the whole sentence. However, the entropy rate for S2 drops to 0 halfway through the sentence. S2 transmits no information after symbol 6. The entropy rate for S1 stays constant throughout.

Inconsistently branching grammars like G2 sanction sentences whose entropy rates fail to be constant. The transmission of information is thus more uneven over the course of the sentence. If entropy rates are uneven, the parser learns little information about the derivation at some points in the sentence and a great deal at other points.

Experiment 2: The Entropy Rates of Simulated Grammars

The simple scenario just described suggests a way to measure the effect of branching inconsistency on entropy rate constancy. Phrases that are mutually inconsistent should be expected to convey a great deal of information at first, then slacken off to convey very little information later. Grammars that contain inconsistent phrases should license sentences whose entropy rates drop precipitously in the middle. If the entropy rates of an inconsistent grammar are averaged over a number of sentences, there should appear a smooth decline in entropy rate from the first sentence symbol to the last. In the following figure, the average entropy rate has a smooth negative slope since each sentence alone begins with a high entropy rate, which then drops decisively.

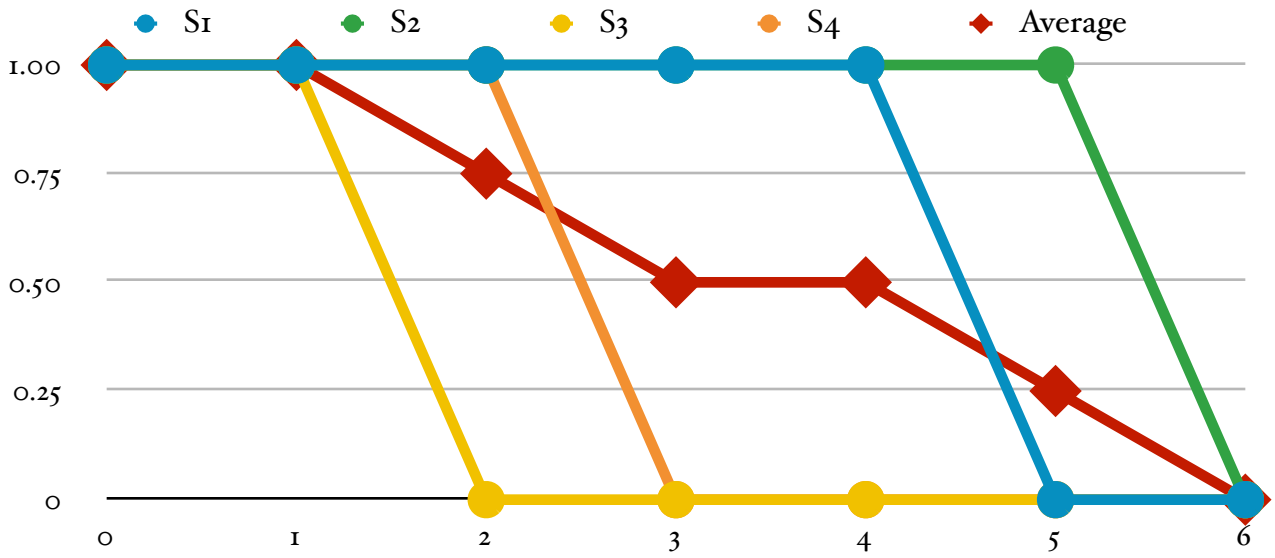


Figure 12: The mean entropy rate for an invented inconsistent grammar shows a smooth, negative slope due to the cumulative effect of sentences whose entropy rates start high and finish low.

On the other hand, grammars that are consistent should license sentences whose entropy rates are more constant. These sentences should average out in a way that preserves the constancy of the entropy rate.

The 32 grammars from Christiansen and Devlin (1997) were used as the basis for a simulation. Each grammar generated 150,000 sentences, and the entropy rates at each sentence position (first, second, third, etc.) were computed by averaging over all sentences that were longer than a given sentence position cardinality. (E.g., only sentences as long as 5 symbols contributed to the entropy rate calculation for symbol 5.) Because of data sparsity for later sentence positions, entropy rates were not calculated past 25 symbols. A linear regression was performed on each grammar, yielding a slope. The slope corresponded to the average change in entropy rate from the beginning to the end of sentences. It was expected that recursively inconsistent grammars would demonstrate more highly negative slopes. To test this, a correlation between slope and inconsistency was performed. "Inconsistency" was here quantified not using Christiansen and Devlin's RRIC but as the raw number of inconsistent, mutually referential phrases (0, 1, or 2).

The correlation between slope and inconsistency was -0.735. Therefore, increased inconsistency resulted in a more negative slope in entropy rate from the beginning of the

sentence to the end of the sentence as expected by our hypothesis. (The entropy rates decreased more over the course of the sentences for inconsistent grammars.) A one-way ANOVA yielded $F(2,29) = 27.185$ with $p < 0.001$.

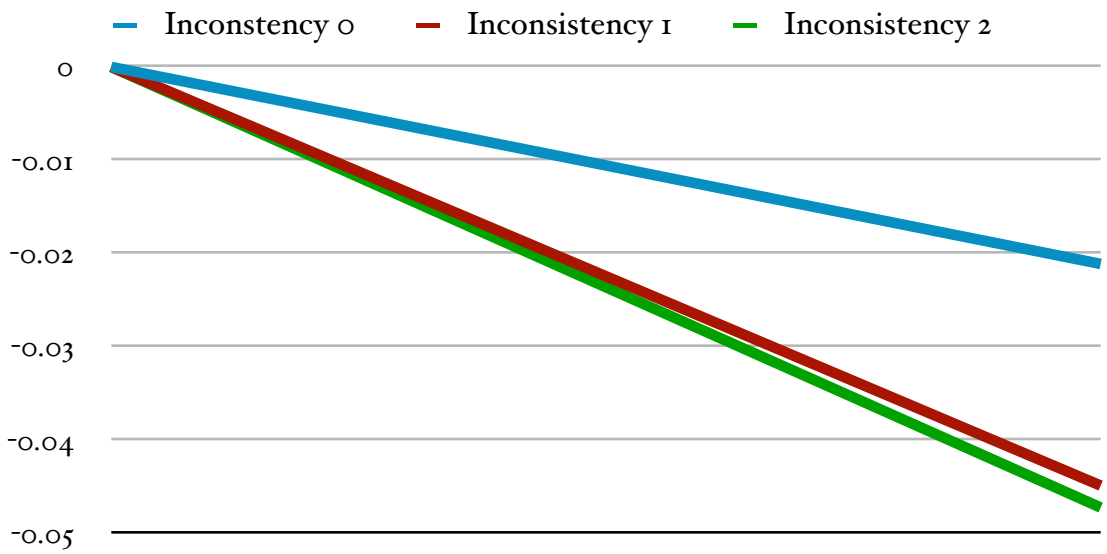


Figure 13: The average regressed entropy rate slope over the course of the sentence for each class of grammars. Consistent grammars show the most constant entropy rates. (The y-axis represents slope.)

Of course, while the analytical argument coupled with the simulations seems to bear out the theory that entropy rate constancy and grammatical consistency are strongly linked phenomena, it is only fair to be cautious about interpreting the results too broadly. Natural languages are a great deal more than syntax, and the information content of a word (in a satisfying theory) is not purely determined by syntactic context. In explaining my results to others, I have frequently struggled with the task of inventing plausible English-language examples that display inconstant entropy rates. It is more or less impossible because words in English are much more than pre-terminal categories. In Christiansen and Devlin's grammar skeleton, there is only one type of adposition; any time the adposition word class is made completely certain by its preceding context, the adposition ceases to communicate any information. In English, even if a preposition's appearance in a sentence can be predicted, there are dozens of different prepositions to choose among. The information content of the word is semantically based and can never vanish.

Thoughts and Conclusions

Roger Levy suggested that one could further test the effects of branching inconsistency by taking English-language corpora for which one has grammars that give reasonably good fits. One could then try reversing various phrase types. Whatever statistic one is interested in could be computed before and after doping the grammar in this manner. Using the techniques described in Hale (2003) to capture per word entropy, one could compute the entropy rates of the grammars in both conditions to see if there is any detectable "real-world" effect. More practically, one could also try artificial language learning experiments with real English words that bind together into meaningful sentences but within foreign grammatical constructs. For example, one could test reading times on sentences composed in Japanese word order against reading times on sentences composed in unattested, inconsistently branching word orders.

Using entropy rate constancy to characterize typological universals is also certainly not limited to branching consistency. It might be possible to explain various grammatical shifts as optimizations of entropy rate constancy, though such analyses could become perilously *post hoc*.

Unresolved in all this discussion is how languages have developed towards consistency. I have attempted to practice unbiased jurisprudence with respect to nature and nurture. Whereas connectionist simulation assumes that nature (the implicit biases of neural networks) has constrained branching inconsistency in natural languages, the principle of entropy rate constancy could be either culturally or biologically ingrained. Genzel and Charniak (2002) note that the speech processing community has researched entropy rate constancy in depth. Among the results, speakers tend to slow down their pronunciation of ambiguous words, which has been construed as a measure to preserve the constancy of entropy rate. These are deliberate acts on the part of the speakers to make themselves better understood. Analogously, it is quite possible that over time people have consciously manipulated the grammatical structure of languages to maintain constant entropy rates in their dialogue.

BIBLIOGRAPHY

Baker, M.C. (2001). *The Atoms of Language*. USA: Basic Books.

Christiansen, M.H. and Devlin, J. (1997). Recursive Inconsistencies are Hard to Learn: A Connectionist Perspective on Universal Word Order Correlations. In M. Shafto and P. Langley (eds.), *Proceedings of the 19th Annual Cognitive Science Society Conference*. Mahwah, NJ: Erlbaum, 113-18.

Christiansen, M.H. and Ellefson, M. (2002). Linguistic Adaptation without Linguistic Constraints: The Role of Sequential Learning in Language Evolution. In Wray, A. (ed.), *The Transition to Language*. Oxford: Oxford University Press, 335-358.

Deacon, T.W. (2003). Universal Grammar and Semiotic Constraints. In Christiansen, M.H. and Kirby, S. (eds.), *Language Evolution*. Oxford: Oxford University Press, 111-139.

Dryer, M.S. (1992). The Greenbergian Word Order Correlations. *Language*, **68**, 81-138.

Elman, J.L. (1991). Finding Structure in Time. *Cognitive Science*, **14**, 179-211.

Gers, F.A. and Schmidhuber, J. (2001). LSTM Recurrent Networks Learn Simple Context-Free and Context-Sensitive Languages. *IEEE Transactions on Neural Networks*, **12**, no. 6., 1333-1340.

Genzel, D. and Charniak, E. (2002). Entropy Rate Constancy in Text. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 199-206

Graves, A., Eck, D., Beringer, N., and Schmidhuber, J. (2004). Biologically Plausible Speech Recognition with LSTM Neural Nets. *Biologically Inspired Approaches to Advanced Information Technology, First International Workshop, BioAdit 2004*. Lausanne, Switzerland.

Greenberg, J.H. (1963). Some Universals of Grammar with Particular Reference to the Order of Meaningful Elements. In Greenberg, J.H. (ed.), *Universals of Language*. London: MIT Press, 73-113.

Hale, J. (2001). A Probabilistic Earley Parser as a Psycholinguistic Model. Proceedings of NAACL-2001.

Hale, J. (2003). The Information Conveyed by Words in Sentences. *Journal of Psycholinguistic Research*, **32**, no. 2, 101-123.

Hawkins, J.A. (1990). A Parsing Theory of Word Order Universals. *Linguistic Inquiry*, **21**, no. 2, 223-261.

Hawkins, J.A. (1994). *A Performance Theory of Order and Consistency*. UK: Cambridge University Press.

Hawkins, J. and Blakeslee, S. (2004). *On Intelligence*. New York: Times Books.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, **9**, 1735-1780.

Kirby, S. (1998). Fitness and the Selective Adaptation of Language. In Hurford, J.R., Studdert-Kennedy, M. and Knight, C. (eds.), *Approaches to the Evolution of Language: Social and Cognitive Bases*. Cambridge: Cambridge University Press.

Kirby, S. and Christiansen, M.H. (2003). From Language Learning to Language Evolution. In Christiansen, M.H. and Kirby, S. (eds.), *Language Evolution*. Oxford: Oxford University Press, 272-294.

Manning, C.D., and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. USA: MIT Press.

Van Everbroeck, E. (2003). Language Type Frequency and Learnability from a Connectionist Perspective. *Linguistic Typology*, **7**, 1-50

The Rosetta Project. (2005). www.rosettaproject.org

Roger Levy, Personal Communication.