ON THE UNSUPERVISED INDUCTION OF PHRASE-STRUCTURE GRAMMARS

CARL DE MARCKEN
MIT Artificial Intelligence Laboratory
545 Technology Square
Cambridge, MA, 02139, USA
cgdemarc@ai.mit.edu

1. Introduction

Researchers investigating the acquisition of phrase-structure grammars from raw text have had only mixed success. In particular, unsupervised learning techniques, such as the inside-outside algorithm (Baker, 1979) for estimating the parameters of stochastic context-free grammars (SCFGs), tend to produce grammars that structure text in ways contrary to our linguistic intuitions. One effective way around this problem is to use hand-structured text like the Penn Treebank (Marcus, 1991) to constrain the learner: (Pereira and Schabes, 1992) demonstrate that the inside-outside algorithm can learn grammars effectively given such constraint, and currently the best performing parsers are trained on treebanks (Black et al., 1992; Magerman, 1995).

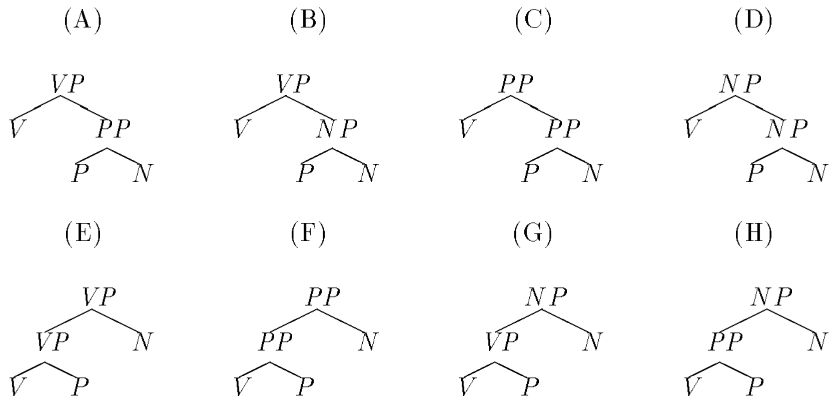
The necessity of bracketed corpora for training is grating to our sensibilities, for several reasons. First, bracketed corpora are not easy to come by. Second, there is a sense that in learning from them, little of interest is going on. In the case of the acquisition of stochastic context-free grammars, the parameters can be read off of a fully-bracketed corpus by simply counting. Finally, the inability of current techniques to learn (without supervision) the parameters we desire suggests that our models and training methods are mismatched to the problem.

This paper examines why some previous approaches have failed to acquire desired grammars without supervision, and proposes that with a different conception of phrase-structure supervision might not be necessary. In particular, it describes in detail some reasons why SCFGs are poor models to use for learning human language, especially when combined with the inside-outside algorithm. Following up on these arguments, it proposes that head-driven grammatical formalisms like link grammars (Sleator and Temperley, 1991) are better suited to the task, and introduces a framework for CFG induction that sidesteps many of the search problems that previous schemes have had. In the end, we hope the analysis presented here convinces others to look carefully at their representations and search strategies before blindly applying them to the language learning task.

We start the discussion by examining the differences between the linguistic and statistical motivations for phrase structure; this frames our subsequent analysis. Then we introduce a simple extension to stochastic context-free grammars, and use this new class of language models in two experiments that pinpoint specific problems with both SCFGs and the search strategies commonly applied to them. Finally, we explore fixes to these problems.

2. Linguistic and Statistical Basis of Phrase Structure

Let us look at a particular example. In English, the word sequence *"walking on ice"* is generally labeled with an internal structure similar to (A).[1]



Why (A) and not one of (B-H)? An introductory linguistics book might proffer the following answers:

[1]We will be deliberately vague about what such dominance and precedence relations represent; obviously different researchers have very different conceptions about the relevence and implications of heirarchical phrase-structure. The specific interpretation given to trees is somewhat irrelevent to our immediate discussion, though various interpretaions will be discussed throughout this paper. In fact, we suspect that for most applications conventional parse trees such as those found in the Penn Treebank, with their historical roots in deletion and substitution phenomena, are a poor choice for a representation.

- *on ice* can move and delete as one unit, whereas *walking on* can not. Thus, *"it is on ice that I walked"* and *"it is walking that I did on ice"* and *"it is ice that I walked on"* are sentences but there in no equivalent form for topicalizing *walking on*. Similarly, *"they walked and jumped on ice"* is grammatical but *"they walked on and jumped on ice"* is awkward. Therefore, if movement and conjunction is of single constituents, phrase-structures (A-D) explain this evidence but (E-H) do not.
- In languages like German where case is overtly manifested in affix and determiner choice, the noun *ice* clearly receives case from the preposition rather than the verb. It seems to make for a simpler theory of language if case is assigned through the government relation, which holds between the preposition and noun in (A-D) but not in (E-H).
- The phrase *walking on ice* acts like a verb: it can conjoin with a verb (*"John walked on ice and sang"*), and takes verbal modifiers (*"John walked on ice slowly"*). So it makes little sense to call it a prepositional phrase or noun phrase, as in (C) or (D). *on ice* does not behave as a noun, so (A) is a better description than (B).

Statistical phrase-structure models of language, such as SCFGs, are motivated by entirely different aspects of language. The measure of merit for a grammar is not how well it explains various structural and interpretive phenomena, but how well it *predicts* the sentences of a corpus. The production rules of a grammar act as a mechanism for specifying statistical dependencies. This suggests that phrase structure can be recovered by grouping sequences of words that occur together more often than independent chance would predict. (Magerman and Marcus, 1990) adopt this approach for parsing sentences, and use a metric based on mutual information between words rather than a traditional grammar to reconstruct phrase-structure. In fact, the heuristic of grouping unusually common sequences lies at the heart of most unsupervised grammar induction mechanisms.

Unfortunately, there is anecdotal and quantitative evidence that simple techniques for estimating context-free grammars by minimizing cross-entropy[2] do not lead to the desired grammars (grammars that agree with structure (A), for instance). (Pereira and Schabes, 1992) explore this topic, demonstrating that an SCFG trained on part-of-speech sequences from English text can have a cross-entropy as low or lower than another but bracket the text much more poorly (tested on hand-annotations). And (Magerman and Marcus, 1990) provide evidence that greedily grouping sequences of words that predict each other is not always a good heuristic;

[2] Readers unfamiliar with the terminology of information theory may wish to consult appendix A for a brief introduction.

they must include in their parsing algorithm a list of sequences (such as noun-preposition) that should not be grouped together in a minimal phrase, in order to prevent their method from mis-bracketing. To understand why, we can look at an example from a slightly different domain.

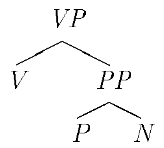
(Olivier, 1968) seeks to acquire a lexicon from unsegmented (spaceless) character sequences by treating each word as a stochastic context-free rule mapping a common nonterminal (call it W) to a sequence of letters; a sentence is a sequence of any number of words and the probability of a sentence is the product over each word of the probability of W expanding to that word. Learning a lexicon consists of finding a grammar that reduces the entropy of a training character sequence. Olivier's learning algorithm soon creates rules such as $W \Rightarrow$ *the* and $W \Rightarrow$ *tobe*. But it also hypothesizes words like *edby*. *edby* is a common English character sequence that occurs in passive constructions like *"the dog was walked by his master"*. Here *-ed* and *by* occur together not because they are part of a common word, but because English syntax and semantics places these two morphemes side-by-side. At a syntactic level, this is exactly why the algorithm of (Magerman and Marcus, 1990) has problems: English places prepositions after nouns not because they are in the same phrase, but because prepositional phrases often adjoin to noun phrases. Any greedy algorithm that builds phrases by grouping units with high mutual information will consequently fail to derive linguistically-plausible phrase structure in many situations.

These results highlight an important point. Anyone who tries to mirror parses found in a treebank with a grammar trained to optimally predict word sequences is relying on a strong assumption, namely that prediction is easiest if it is based on a linguist's conception of phrase structure. With the wrong class of language models, this assumption is obviously false. For example, the maximum-likelihood distribution for any set of n sentences is the one that assigns probability $\frac{1}{n}$ to each of these sentences and 0 to every other. A SCFG with n rules that each produce a single sentence captures this distribution, but provides no information about linguistic structure whatsoever. Plainly, either this is too unconstrained a class of models or the evaluation metric is incorrect (probably both). But, even with a more constrained class of models, *the success of unsupervised, statistical grammar induction is intimately tied to how models take advantage of linguistic structure.* The next section explores this issue in greater depth.

3. A Simple Language Model

The preceeding arguments might lead us to believe that basing phrase-structure grammar induction on minimization of cross-entropy is a poor idea. However, in this paper we will not discuss whether statistical optimization is the proper way to view language acquisition: our goal is only to better understand why current statistical methods produce the "wrong" answer and to explore ways of fixing them. With an eye towards this, we extend the class of stochastic context-free grammars with the notion of a head. As we will see, in this extended class of grammars, there is reason to believe that the "linguistically plausible" members are also the ones with the lowest cross-entropy. This will enable us to pinpoint some failures in both the grammatical representation and the induction scheme.

Let us look again at (A), reproduced below, and center discussion on a class of models in which a binary context-free rule $Z \Rightarrow XY$ with terminals X and Y generates a symbol $x \in \mathcal{X}$ from the distribution $p_X(\cdot)$ and another symbol $y \in \mathcal{Y}$ from the distribution $p_{Y|X}(\cdot, x)$.[3] Given this formulation, the joint entropy of the sequence XY is $H(X) + H(Y|X) = H(X) + H(Y) - I(X,Y)$. The point here is that using such a context-free rule to model a sequence of two words reduces the entropy of the language from a model that treats the two words as independent, by precisely the mutual information between the two words.

(A) 

In English, verbs and prepositions in configuration (A) are closely coupled semantically, probably more closely than prepositions and nouns, and we would expect that the mutual information between the verb and preposition would be greater than between the preposition and noun, and greater still than between the verb and the noun.[4]

$$I(V,P) > I(P,N) > I(V,N)$$

Under this class of models, structure (A) has entropy $H(V) + H(P) + H(N|P) = H(V)+H(P)+H(N)-I(P,N)$, which is higher than the entropy of structures (E-H), $H(V)+H(P)+H(N)-I(V,P)$, and we wouldn't expect a learning mechanism based on such a class of models to settle on (A).

However, this simple class of models only captures relations between adjacent words within the same minimal phrase. In (A), it completely ignores the relation between the verb and the prepositional phrase, save to

[3] Here we are mixing notation somewhat. X and Y are playing the roles of parts of speech, treated as random variables. Particular instances of the random variables, such as x, play the role of a word. For further explanation of notation, see appendix A.

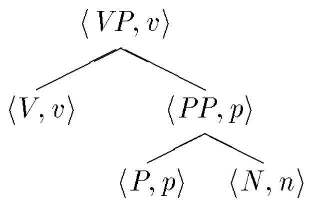
[4] The small size of the set of prepositions imposes an upper bound on $I(V,P)$ and $I(P,N)$, so it may be that $I(V,N) > I(V,P) > I(P,N)$ in some circumstances, but this point is not worth dwelling on here; in section 5.2 we will expand on it.

predict that a prepositional phrase (*any* prepositional phrase) will follow the verb. We again extend the class, specifying that nonterminals exhibit the distributional properties of their heads. We will write a phrase Z that is headed by a word z as $\langle Z, z\rangle$. Each grammar rule will look like either $\langle Z', z\rangle \Rightarrow \langle Z, z\rangle\langle Y, y\rangle$ or $\langle Z', z\rangle \Rightarrow \langle Y, y\rangle\langle Z, z\rangle$ (abbreviated $Z' \Rightarrow ZY$ and $Z' \Rightarrow YZ$) and the probability model is

$$\begin{aligned} p(\langle Z, z\rangle\langle Y, y\rangle | \langle Z', z'\rangle, Z' \Rightarrow ZY) &= p_{Z|Z'}(z, z') \cdot p_{Y|Z}(y, z) \\ &= \delta(z, z') \cdot p_{Y|Z}(y, z). \end{aligned} \tag{1}$$

$$\begin{aligned} p(\langle Y, y\rangle\langle Z, z\rangle | \langle Z', z'\rangle, Z' \Rightarrow YZ) &= p_{Z|Z'}(z, z') \cdot p_{Y|Z}(y, z) \\ &= \delta(z, z') \cdot p_{Y|Z}(y, z). \end{aligned} \tag{2}$$

Of course, this class of models is strongly equivalent to ordinary context-free grammars. We could substitute, for every rule $Z' \Rightarrow ZY$, a large number of word-specific rules $\langle Z', z_i\rangle \Rightarrow \langle Z, z_i\rangle\langle Y, y_j\rangle$ with probabilities $p(Z' \Rightarrow ZY) \cdot p_{Y|Z}(y_j, z_i)$. Using this new formalism, the head properties of (A) look like



and the entropy is

$$H(V) + H(P) + H(N) - I(V, P) - I(P, N).$$

The grammar derived from (A) is optimal under this class of models though (C), (F), and (H) are equally good. They could be distinguished from (A) in longer sentences because they pass different head information out of the phrase. In fact, the grammar model derived from (A) is as good as any possible model that does not condition N on V. Under this class of models there is no benefit to grouping two words with high mutual information together in the same minimal phrase; it is sufficient for both to be the heads of phrases that are adjacent at some level.

Of course, we are not claiming this class of models is sufficient to bring the statisticians' and linguists' views of language into perfect alignment. But it illustrates how, by allowing a statistical model to take advantage of the structure it imposes on a sentence, one can hope for a happy synergy. We can ask whether our parameter estimation algorithms are adequate for learning with this class of grammars, and whether the class itself still needs improving, two questions answered by the experiments described next.

4. Two Experiments

We have built a feature-based Earley parser for stochastic grammars that can be trained using the inside-outside algorithm. Here we describe two tests that explore the interaction of the head-driven language models described above with this parser and training method.

For all the tests presented here, grammars are learned by starting with an exhaustive set of stochastic context-free rules of a certain form. Rule probabilities are then estimated from a test corpus. This is the same general procedure as used by (Lari and Young, 1990; Briscoe and Waegner, 1992; Pereira and Schabes, 1992) and others. For parts-of-speech Y and Z, the rules in the base grammar are

$$\begin{array}{ll} S \Rightarrow ZP & ZP \Rightarrow Z\ YP \\ ZP \Rightarrow ZP\ YP & ZP \Rightarrow YP\ Z \\ ZP \Rightarrow YP\ ZP & ZP \Rightarrow Z \end{array}$$

where S is the root nonterminal. As is ususal with stochastic context-free grammars, every rule has an associated probability, and the probabilities of all the rules that expand a single nonterminal sum to one. Furthermore, each word and phrase has an associated head word (represented as a feature value that is propagated from the Z or ZP on the right hand side of the above rules to the left hand side). The parser is given the part of speech of each word.

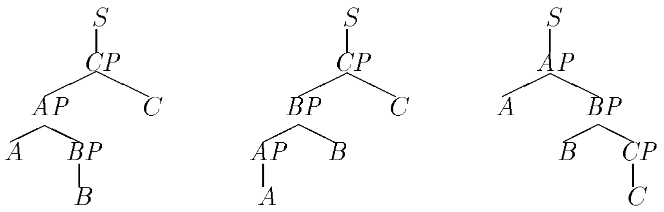
For binary rules, as per equations (1) and (2), the distribution of the non-head word is conditioned on the head (similarly to a bigram model). Initially, all word bigrams are initialized to uniform distributions, and context-free rule probabilities are initialized to a small random perturbation of a uniform distribution.

4.1. SEARCH PROBLEMS FOR A SIMPLE SENTENCE

We created a test corpus of 1000 sentences, each 3 words long with a constant part-of-speech pattern ABC. Using 8 equally probable words per part-of-speech, we chose a word distribution over the sentences with the following characteristics:

$$\begin{array}{rcl} I(A,B) & = & 1 \text{ bit.} \\ I(B,C) & = & 0.19 \text{ bits.} \\ I(A,C) & = & 0 \text{ bits.} \end{array}$$

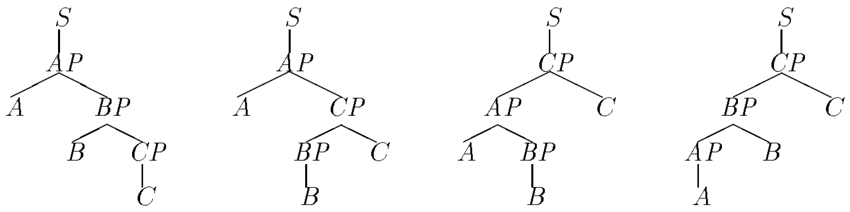
In other words, given knowledge of the first word in the sentence, predicting the second word is as difficult as guessing between four equally-likely words, and knowing the second word makes predicting the third about as difficult as guessing between seven words. Knowing the first gives no information about the third.[5] This is qualitatively similar to the distribution we assumed for verbs, nouns, and prepositions in configuration (A), and has entropy $3 + (3 - 1) + (3 - .19) = 7.81$ bits. Across 20 runs, the training algorithm converged to three different grammars:[6]



12 times	2 times	6 times
$\hat{H} = 7.94$	$\hat{H} = 7.71$	$\hat{H} = 7.71$

One fact is immediately striking: even with such simple sentences and rule sets, more often than not the inside-outside algorithm converges to a suboptimal grammar. To understand why, let us ignore recursive rules ($ZP \Rightarrow ZP\ YP$) for the moment. Then there are four possible parses of ABC (cross-entropy with source given below- lower is better model):

(I) (J) (K) (L)



(I)	(J)	(K)	(L)
$\hat{H} = 7.81$	$\hat{H} = 8.81$	$\hat{H} = 8.00$	$\hat{H} = 7.81$

[5] Such distributions are not difficult to create. If a word is represented by a binary vector $b_0 b_1 b_2$, then a distribution with $I(A, B) = 1$, $I(B, C) = 1$, and $I(A, C) = 0$ results from enforcing $b_0(A) = b_0(B)$ and $b_1(B) = b_1(C)$ on an otherwise uniform distribution.

[6] That is to say, after the cross-entropy had ceased to decrease on a given run, the parser settled on one of these structures as the Viterbi parse of each sentences in the corpus. The cross-entropy of the two best grammars is lower than the source entropy because the corpus is finite and randomly generated, and has been be overfitted.

During the first pass of the inside-outside algorithm, assuming near-uniform initial rule probabilities, each of these parses will have equal posterior probabilities. They are equally probable because they use the same number of expansions[7] and because word bigrams are uniform at the start of the parsing process. Thus, the estimated probability of a rule after the first pass is directly proportional to how many of these parse trees the rule participates in. The rules that occur more than one time are

$$\begin{array}{ll} AP \Rightarrow A\ BP & \text{(parses I,K)} \\ CP \Rightarrow BP\ C & \text{(parses J,L)} \\ BP \Rightarrow B & \text{(parses J,K).} \end{array}$$

Therefore, on the second iteration, these three rules will have higher probabilities than the others and will cause parses (J) and (K) to be favored over (I) and (L) (with (K) favored over (J) because $I(A,B) + I(A,C) > I(B,C) + I(A,C)$). It is to be expected then, that the inside-outside algorithm favors the suboptimal parse (K): at its start the inside-outside algorithm is guided by tree counting arguments, not mutual information between words. This suggests that the inside-outside algorithm is likely to be highly sensitive to the form of grammar and how many different analyses it permits of a sentence.

Why, later, does the algorithm not move towards a global optimum? The answer is that the inside-outside algorithm is supremely unsuited to learning with this representation. To understand this, notice that to move from the initially favored parse (K) to one of the optimal ones (I) and (L), three nonterminals must have their most probable rules switched:

(K)	$\longrightarrow$	(L)
$AP \Rightarrow A\ BP$	$\longrightarrow$	$AP \Rightarrow A$
$BP \Rightarrow B$	$\longrightarrow$	$BP \Rightarrow AP\ B$
$CP \Rightarrow AP\ C$	$\longrightarrow$	$CP \Rightarrow BP\ C$

To simplify the present analysis, let us assume the probability of $S \Rightarrow CP$ is held constant at 1, and that the rules not listed above have probability 0. In this case, we can write the probabilities of the left three rules as q_{A}, q_{B} and q_{C} and the probabilities of the right three rules as $\overline{q}_{\text{A}} = 1 - q_{\text{A}}$, $\overline{q}_{\text{B}} = 1 - q_{\text{B}}$ and $\overline{q}_{\text{C}} = 1 - q_{\text{C}}$. Now, for a given sentence *abc* there are only two parses with non-zero probabilities, (K) and (L). The prior probability of parse (K) is $q_{\text{A}}q_{\text{B}}q_{\text{C}}$ and the prior probability of parse (L) is $\overline{q}_{\text{A}}\overline{q}_{\text{B}}\overline{q}_{\text{C}}$.

[7]This is why we can safely ignore recursive rules in this discussion. Any parse that involves one will have a bigger tree and be significantly less probable.

The probability of abc given (K) is $p_{A|C}(a,c)p_{B|A}(b,a)$ and given (L) is $p_C(c)p_{B|C}(b,c)p_{A|B}(a,b)$. Thus, the posterior probability of parse (K) is[8]

$$\begin{aligned} p(\mathrm{K}|abc) &= \frac{p(K,abc)}{p(K,abc)+p(L,abc)} = \frac{1}{1+\frac{p(L,abc)}{p(K,abc)}} \\ &= \frac{1}{1+\frac{\overline{q}_{\mathrm{A}}\overline{q}_{\mathrm{B}}\overline{q}_{\mathrm{C}}p_{B|C}(b,c)p_{A|B}(a,b)}{q_{\mathrm{A}}q_{\mathrm{B}}q_{\mathrm{C}}p_{A|C}(a,c)p_{B|A}(b,a)}} \\ &= \frac{1}{1+\frac{\overline{q}_{\mathrm{A}}\overline{q}_{\mathrm{B}}\overline{q}_{\mathrm{C}}p_{C|B}(c,b)}{q_{\mathrm{A}}q_{\mathrm{B}}q_{\mathrm{C}}p_{C|A}(c,a)}}. \end{aligned}$$

Since the inside-outside algorithm reestimates q_{A}, q_{B} and q_{C} directly from the sums of the posterior probabilities of (K) and (L) over the corpus, the probability update rule from one iteration to the next is well approximated by

$$q_{\mathrm{A}}, q_{\mathrm{B}}, q_{\mathrm{C}} \leftarrow \frac{1}{1+\frac{\overline{q}_{\mathrm{A}}\overline{q}_{\mathrm{B}}\overline{q}_{\mathrm{C}}}{q_{\mathrm{A}}q_{\mathrm{B}}q_{\mathrm{C}}}\alpha}.$$

where α is the expected value of $p_{C|B}(c,b)/p_{C|A}(c,a)$ over the training corpus, about $\frac{8}{7}$ in the above test. Figure 4.1 graphically depicts the evolution of this dynamical system. What is striking in this figure is that the inside-outside algorithm is so attracted to grammars whose terminals concentrate probability on small numbers of rules that it is incapable of performing real search. Instead, it zeros in on the nearest such grammar, only biased slightly by its relative merits. We now have an explanation for why the inside-outside algorithm converges to the suboptimal parse (K) so often: the first ignorant iteration of the algorithm biases the parameters towards (K), and subsequently there is an overwhelming tendency to move to the nearest deterministic grammar. This is a strong indication that the algorithm is a poor choice for estimating the parameters of grammars that have competing rule hypotheses.

4.2. MULTIPLE EXPANSIONS OF A NONTERMINAL

For this test, the sentences were four words long ($ABCD$), and we chose a word distribution with the following characteristics:

[8] In the following derivation, understand that for word bigrams $p_{A|B}(a,b) = p_{B|A}(b,a)$ because $p_A(a) = p_B(b) = \frac{1}{8}$.

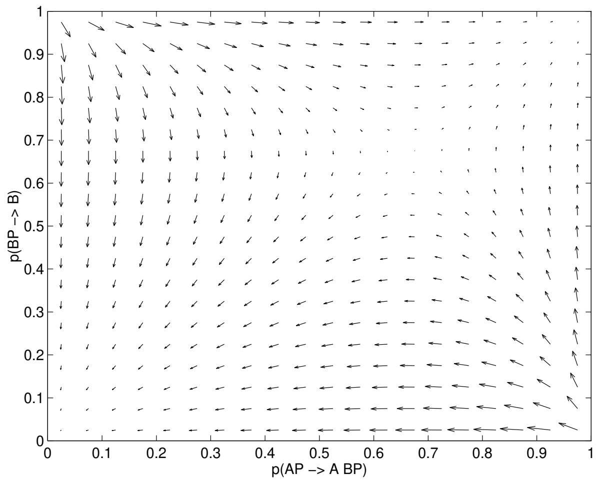
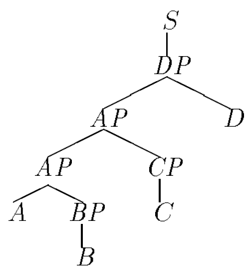
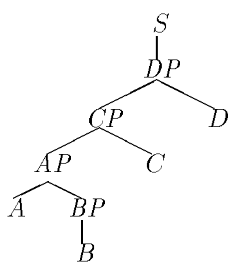


Figure 4.1: The dynamical properties of the inside-outside algorithm. The x-axis is q_{A} and the y-axis is q_{B}. The vectors represent the motion of the parameters from one iteration to the next when $\alpha = 2$ and $q_{\mathrm{C}} = .5$. Notice that the upper right corner (grammar K) and the lower left (grammar L) are stationary points (local maxima), and that the region of attraction for the global optimum (L) is bigger than for (K), but that there is still a very substantial set of starting points from which the algorithm will converge to the suboptimal grammar. $\alpha = 2$ is plotted instead of $\alpha = \frac{8}{7}$ because this better depicts the asymmetry mutual information between words introduces; with $\alpha = \frac{8}{7}$ the two regions of attraction would be of almost equal area.

$$\begin{array}{ll} I(A,B) = 1 \text{ bit.} & I(B,C) = 0 \text{ bits.} \\ I(A,C) = 1 \text{ bit.} & I(B,D) = 0 \text{ bits.} \\ I(A,D) = 1 \text{ bit.} & I(C,D) = 0 \text{ bits.} \end{array}$$

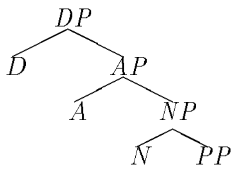
It might seem that the grammar (M) is a minimal-entropy grammar for this corpus

(M) 

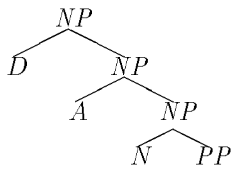
(N) 

since it makes the head A available to predict B, C, and D. Without multiple expansions rules for AP, it is impossible for A to enter into this many head relationships. But the gain of one bit in word prediction is offset by a loss of at least two bits from uncertainty in the expansion of AP. Even if $p(AP \Rightarrow A\ BP) = p(AP \Rightarrow AP\ CP) = 1/2$, the probability of the structure $ABCD$ under the above grammar is one-quarter that assigned by a grammar with no expansion ambiguity. So, the grammar (N) assigns higher probabilities to the corpus, even though it fails to model the dependency between A and D. This is a general problem with SCFGs: there is no way to optimally model multiple ordered adjunction without increasing the number of nonterminals. Not surprisingly, the learning algorithm never converges to the recursive grammar during test runs on this corpus. Others have noted the inadequacy of modeling nonterminal expansion as an independent process; *history-based grammars* (Black et al., 1992; Magerman, 1995) are one response, but unfortunately one poorly suited to unsupervised learning.

What broader implication does this deficiency of SCFGs have for context-free grammar based language acquisition? It suggests if we were to estimate a grammar from English text, that the sequence complex noun phrase surface form $D\ A\ N\ PP$ is far more likely to get the interpretation



than



and therefore that, for many subject and object noun phrases, the noun will never enter into a bigram relationship with the verb. Obviously sufficient mutual information between nouns and verbs, adjectives, and determiners would force the global optimum to include multiple expansions of the NP category, but it seems likely (given the characteristics of the inside-outside algorithm) that before such mutual information could be inferred from text, the search process would settle on a local optimum that does not pass the noun feature out. This case is another illustration of how intimately tied the form of stochastic grammars is to their ability to reproduce "linguistically plausible" structure in an unsupervised framework.

5. Attacking the Problems

We have argued that the grammatical representations commonly used for unsupervised learning will never converge on linguistically plausible structures, both because they fail to acknowledge the linguistic basis of phrase structure, and because the search procedures associated with them tend to get stuck in local optima. Although they will not be fleshed out in great detail here, we present sketches of "fixes" for some of the problems our analysis has touched on. These are essentially proposals for further research, and are offered to show that there is still hope unsupervised techniques *can* be made to work for grammar induction.

5.1. RULE INTERACTION AND LINK GRAMMARS

In the first experiment described above, the failure of the inside-outside algorithm to converge to the optimal grammar is due to the discontinuous nature of the search space, a consequence of rule interaction. Three different parameters are tightly coupled, and none can be determined independently of the rest. In this case, the space is more complicated than it need be, because nonterminals are labeled.

Fortunately, the space can be flattened. In particular, grammars can be represented in terms of head relations, in a manner very similar to the link grammars of (Sleator and Temperley, 1991). Let us look again the sequence VPN. There are only three words here, and therefore three heads. Assuming a head-driven bigram model as before, there are only three possible anlayses of this sequence, which we write by listing the pairs of words that enter into predictive relationships:

Head Relations	Equivalent Parse Trees
$V-P, V-N$	E,G
$V-P, P-N$	A,C,F,H
$V-N, P-N$	B,D

To map back into traditional phrase structure grammars, linking two heads $X-Y$ is the same as specifying that there is some phrase XP headed by X which is a sibling to some phrase YP headed by Y. Of course, using this representation all of the optimal phrase structure grammars (A,C,F and H) are identical. Thus we have a representation which has factored out many details of phrase structure that are unimportant as far as minimizing entropy is concerned.

Simplifying the search space reaps additional benefits. A greedy approach to grammar acquisition that iteratively hypothesizes relations between the words with highest mutual information will first link V to P, then P to N, producing exactly the desired result for this example. And the distance in parse or grammar space between competing proposals is at most one relation (switching $V-P$ to $V-N$, for instance), whereas three different rule probabilities may need to be changed in the SCFG representation. This suggests that learning algorithms based on this representation are far less likely to encounter local optima. Finally, since what would have been multiple parse hypotheses are now one, a Viterbi learning scheme is more likely to estimate accurate counts. This is important, given the computational complexity of estimating long-distance word-pair probabilities from unbracketed corpora.

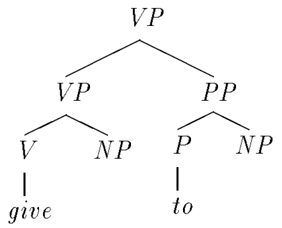
5.2. "FRINGE" RULES AND LEARNING

The naive assumption that nonterminal expansions are statistically independent causes many problems for statistical induction algorithms, as we have seen. One obvious quick-fix is to permit non-binary production rules. For instance, rather than associate a binary-branching structure with a complex noun phrase, it could be modeled with a single rule $NP \Rightarrow D\ A\ N\ PP$. These complex rules are a more natural representation for idiomatic sequences like *for the last time*, where it makes little sense to treat the four words as a chain of pairwise relationships. But there are also many good reasons not to use such rules:

1. There is a much greater risk of overtraining; the increased number of parameters makes the estimation of all of them less reliable.
2. Since the number of possible rules is enormous, to be practical some incremental means of hypothesizing rules must be incorporated into the learning algorithm.
3. If lengthy rules are used, parses will contain very little internal structure, and fail to explain the grammatical regularities that occur even within idiomatic passages.

The first and second point are partially addressed in the schemes of (Stolcke, 1994) and (Chen, 1995), who use a minimum description length (MDL) criterion to reduce the risk of over-parameterization, and incorporate mechanisms for incrementally adding and deleting rules from the grammar. But their schemes fare poorly on the last point, and (being incremental and greedy) are subject to the search problems discussed in section 2.

But notice that complex rules may be decomposable into simple ones. In particular, the right hand side of a rule looks very much like a sentence (one that may contain nonterminals as well as terminals), and therefore it makes sense to treat it as the fringe of a partial derivation tree. For instance, the rule $VP \Rightarrow$ *give* NP *to* NP is the fringe of the tree



and can be represented by the left derivation string

$$VP \Rightarrow VP\ PP \| VP \Rightarrow V\ \ NP \| V \Rightarrow give \| \diamond \| PP \Rightarrow P\ \ NP \| P \Rightarrow to \| \diamond$$

where the symbol $\diamond$ indicates that a nonterminal is not expanded. Notice several consequences of thinking about rules in this way:

- The probability of a rule can be computed in (almost) the same way as the probability of a sentence, and therefore the cost of representing the grammar is easily incorporated into a minimum description length formulation.
- The optimal representation of a production rule can be computed using standard parsing techniques.
- Because complex rules are represented in terms of simpler ones, they have an implicit internal structure (each rule is a tree), which can be reconstructed to give detailed structure even to sentences parsed with long, flat rules.
- The notion of concatenating derivation strings leads naturally to a scheme for hypothesizing new production rules.

Furthermore, because the useful information in a rule is contained on its surface (rather than in its representation), the representation can be continually recomputed during the search process. Therefore, even if a rule $VP \Rightarrow$ *walk on* is created during the learning process, as soon as some mechanism combines it with the word *water* to produce $VP \Rightarrow$ *walk on water*, the final sequence can be reanalyzed into a $[_{VP}\ V\ [_{PP}\ P\ N]]$ structure. In fact, given competition from $VP \Rightarrow$ *walk on water*, we would expect the original rule to be only rarely applied, and it could be deleted. Thus, many of the search problems associated with greedy strategies disappear, because the history of the search process plays little role in the structure assigned to a sentence. In a sense, the state of the search algorithm is no longer a grammar, but a set of grammatical constructs (idioms, phrases, etc). It is in finding a compact representation for these constructs that traditional phrase structure emerges. More details regarding this approach can be found in (de Marcken, 1995).

6. Conclusions

This paper has presented a detailed analysis of why naive unsupervised grammar induction schemes do not reproduce linguistic structure when applied to raw text, and has suggested that new grammatical representations and search algorithms may improve performance dramatically. We hope that this study convinces others to look carefully at their representations and search strategies before blindly applying them to language, and motivates researchers to study the relationship between linguistic theory and their learning framework.

References

James K. Baker. 1979. Trainable grammars for speech recognition. In *Proceedings of the 97th Meeting of the Acoustical Society of America*, pages 547–550.

E. Black, F. Jelinek, J. Lafferty, D. M. Magerman, R. Mercer, and S. Roukos. 1992. Toward history-based grammars: Using richer models for probabilistic parsing. In *Proceedings of the February 1992 DARPA Speech and Natural Language Workshop.*

Ted Briscoe and Nick Waegner. 1992. Robust stochastic parsing using the inside-outside algorithm. In *Proc. of the AAAI Workshop on Probabilistic-Based Natural Language Processing Techniques*, pages 39–52.

Stanley F. Chen. 1995. Bayesian grammar induction for language modeling. In *Proc. 32nd Annual Meeting of the Association for Computational Linguistics*, pages 228–235, Cambridge, Massachusetts.

Thomas M. Cover and Joy A. Thomas. 1991. *Elements of Information Theory.* John Wiley & Sons, New York, NY.

Carl de Marcken. 1995. The unsupervised acquisition of a lexicon from continuous speech. Memo A.I. Memo 1558, MIT Artificial Intelligence Lab., Cambridge, Massachusetts.

K. Lari and S. J. Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–56.

David M. Magerman and Mitchell P. Marcus. 1990. Parsing a natural language using mutual information statistics. In *Proc. of the American Association for Artificial Intelligence*, pages 984–989.

David M. Magerman. 1995. Statistical decision-tree models for parsing. In *Proc. 32nd Annual Meeting of the Association for Computational Linguistics*, pages 276–283, Cambridge, Massachusetts.

Mitchell Marcus. 1991. Very large annotated databse of American English. In *Proceedings of the DARPA Speech and Natural Language Workshop.*

Donald Cort Olivier. 1968. *Stochastic Grammars and Language Acquisition Mechanisms.* Ph.D. thesis, Harvard University, Cambridge, Massachusetts.

Fernando Pereira and Yves Schabes. 1992. Inside-outside reestimation from partially bracketed corpora. In *Proc. 29th Annual Meeting of the Association for Computational Linguistics*, pages 128–135, Berkeley, California.

Daniel D. K. Sleator and Davy Temperley. 1991. Parsing english with a link grammar. Technical Report CMU-CS-91-196, Carnegie Mellon University, Pittsburgh, Pennsylvania.

Andreas Stolcke. 1994. *Bayesian Learning of Probabalistic Language Models.* Ph.D. thesis, University of California at Berkeley, Berkeley, CA.

A. Definitions

The *entropy* of a discrete random variable A with domain $\mathcal{A}$, denoted $H(A)$ is defined by

$$H(A) \equiv \sum_a -p_A(a) \log p_A(a).$$

Intuitively, the entropy of a random variable is a measure (in *bits*, if the logarithm base is 2) of the uncertainty in the variable's distribution. For a Bernoulli event such as a coin toss (with probability of heads q), the entropy is $-q \log q - (1-q)\log(1-q)$. This achieves a maximum of 1 bit at the least predictable point $q = \frac{1}{2}$, and a minimum of 0 bits at $q = 0$ and $q = 1$, where the outcome is a certainty. Entropy is always bounded below by 0 and above by $\log |\mathcal{A}|$. The *cross-entropy* between two distributions $p_A(\cdot)$ and $p_B(\cdot)$, defined by

$$\hat{H}_A(B) \equiv \sum_a -p_A(a) \log p_B(a),$$

is a measure of how well the distribution $p_B(\cdot)$ predicts A. For example, if the distribution $p_A(\cdot)$ is over sentences in a corpus and B is a sentence produced by a stochastic grammar, then $\hat{H}_A(B)$ is a measure of how well the grammar models the corpus. The cross-entropy achieves a minimum of $H(A)$ when the distributions of A and B are identical. The *joint entropy* and *conditional entropy* of two random variables A and B are defined by

$$H(A, B) \equiv \sum_{a,b} -p_{A,B}(a,b) \log p_{A,B}(a,b).$$

$$H(A|B) \equiv \sum_{a,b} -p_{A,B}(a,b) \log p_{A|B}(a,b).$$

Thus $H(A, B)$ measures the uncertainty of the joint distribution of A and B, and $H(A|B)$ measures of the uncertainty of A given knowledge of B. Conveniently, $H(A, B) = H(A) + H(B|A) = H(B) + H(A|B)$. Finally, the *mutual information* between A and B,

$$\begin{aligned} I(A,B) &\equiv \sum_{a,b} p_{A,B}(a,b) \log \frac{p_{A,B}(a,b)}{p_A(a)p_B(b)} \\ &= H(A) - H(A|B) \\ &= H(B) - H(B|A) \end{aligned}$$

is a measure of the dependence between A and B. It is zero if and only if A and B are independent, and is bounded above by $\min(H(A), H(B))$. For a deeper introduction to these terms from information theory, see (Cover and Thomas, 1991).

INDEX