INSIDE-OUTSIDE REESTIMATION FROM PARTIALLY BRACKETED CORPORA

Fernando Pereira
2D-447, AT&T Bell Laboratories
PO Box 636, 600 Mountain Ave
Murray Hill, NJ 07974-0636
pereira@research.att.com

Yves Schabes
Dept. of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104-6389
schabes@unagi.cis.upenn.edu

ABSTRACT

The inside-outside algorithm for inferring the parameters of a stochastic context-free grammar is extended to take advantage of constituent information (constituent bracketing) in a partially parsed corpus. Experiments on formal and natural language parsed corpora show that the new algorithm can achieve faster convergence and better modeling of hierarchical structure than the original one. In particular, over 90% test set bracketing accuracy was achieved for grammars inferred by our algorithm from a training set of hand-parsed part-of-speech strings for sentences in the Air Travel Information System spoken language corpus. Finally, the new algorithm has better time complexity than the original one when sufficient bracketing is provided.

1. MOTIVATION

The most successful stochastic language models have been based on finite-state descriptions such as n-grams or hidden Markov models (HMMs) (Jelinek et al., 1992). However, finite-state models cannot represent the hierarchical structure of natural language and are thus ill-suited to tasks in which that structure is essential, such as language understanding or translation. It is then natural to consider stochastic versions of more powerful grammar formalisms and their grammatical inference problems. For instance, Baker (1979) generalized the parameter estimation methods for HMMs to stochastic context-free grammars (SCFGs) (Booth, 1969) as the inside-outside algorithm. Unfortunately, the application of SCFGs and the original inside-outside algorithm to natural-language modeling has been so far inconclusive (Lari and Young, 1990; Jelinek et al., 1990; Lari and Young, 1991).

Several reasons can be adduced for the difficulties. First, each iteration of the inside-outside algorithm on a grammar with n nonterminals may require $O(n^3|w|^3)$ time per training sentence w, while each iteration of its finite-state counterpart training an HMM with s states requires at worst $O(s^2|w|)$ time per training sentence. That complexity makes the training of sufficiently large grammars computationally impractical.

Second, the convergence properties of the algorithm sharply deteriorate as the number of nonterminal symbols increases. This fact can be intuitively understood by observing that the algorithm searches for the maximum of a function whose number of local maxima grows with the number of nonterminals. Finally, while SCFGs do provide a hierarchical model of the language, that structure is undetermined by raw text and only by chance will the inferred grammar agree with qualitative linguistic judgments of sentence structure. For example, since in English texts pronouns are very likely to immediately precede a verb, a grammar inferred from raw text will tend to make a constituent of a subject pronoun and the following verb.

We describe here an extension of the inside-outside algorithm that infers the parameters of a stochastic context-free grammar from a partially parsed corpus, thus providing a tighter connection between the hierarchical structure of the inferred SCFG and that of the training corpus. The algorithm takes advantage of whatever constituent information is provided by the training corpus bracketing, ranging from a complete constituent analysis of the training sentences to the unparsed corpus used for the original inside-outside algorithm. In the latter case, the new algorithm reduces to the original one.

Using a partially parsed corpus has several advantages. First, the the result grammars yield constituent boundaries that cannot be inferred from raw text. In addition, the number of iterations needed to reach a good grammar can be reduced; in extreme cases, a good solution is found from parsed text but not from raw text. Finally, the

new algorithm has better time complexity when sufficient bracketing information is provided.

2. PARTIALLY BRACKETED TEXT

Informally, a partially bracketed corpus is a set of sentences annotated with parentheses marking constituent boundaries that any analysis of the corpus should respect. More precisely, we start from a corpus C consisting of ***bracketed strings***, which are pairs $c = (w, \mathcal{B})$ where w is a string and $\mathcal{B}$ is a ***bracketing*** of w. For convenience, we will define the length of the bracketed string c by $|c| = |w|$.

Given a string $w = w_1 \cdots w_{|w|}$, a ***span*** of w is a pair of integers (i, j) with $0 \leq i < j \leq |w|$, which delimits a substring ${}_iw_j = w_{i+1} \cdots w_j$ of w. The abbreviation ${}_iw$ will stand for ${}_iw_{|w|}$.

A bracketing $\mathcal{B}$ of a string w is a finite set of spans on w (that is, a finite set of pairs or integers (i, j) with $0 \leq i < j \leq |w|$) satisfying a consistency condition that ensures that each span (i, j) can be seen as delimiting a string ${}_iw_j$ consisting of a sequence of one of more. The consistency condition is simply that no two spans in a bracketing may *overlap*, where two spans (i, j) and (k, l) overlap if either $i < k < j < l$ or $k < i < l < j$.

Two bracketings of the same string are said to be *compatible* if their union is consistent. A span s is *valid* for a bracketing $\mathcal{B}$ if $\{s\}$ is compatible with $\mathcal{B}$.

Note that there is no requirement that a bracketing of w describe fully a constituent structure of w. In fact, some or all sentences in a corpus may have empty bracketings, in which case the new algorithm behaves like the original one.

To present the notion of compatibility between a derivation and a bracketed string, we need first to define the ***span*** of a symbol occurrence in a context-free derivation. Let $(w, \mathcal{B})$ be a bracketed string, and $\alpha_0 \Rightarrow \alpha_1 \Rightarrow \cdots \Rightarrow \alpha_m = w$ be a derivation of w for (S)CFG G. The span of a symbol occurrence in α_j is defined inductively as follows:

- If $j = m$, $\alpha_j = w \in \Sigma^*$, and the span of w_i in α_j is $(i - 1, i)$.
- If $j < m$, then $\alpha_j = \beta A \gamma$, $\alpha_{j+1} = \beta X_1 \cdots X_k \gamma$, where $A \rightarrow X_1 \cdots X_k$ is a rule of G. Then the span of A in α_j is (i_1, j_k), where for each $1 \leq l \leq k$, (i_l, j_l) is the span of X_l in α_{j+1}. The spans in α_j of the symbol occurrences in β and γ are the same as those of the corresponding symbols in α_{j+1}.

A derivation of w is then compatible with a bracketing $\mathcal{B}$ of w if the span of every symbol occurrence in the derivation is valid in $\mathcal{B}$.

3. GRAMMAR REESTIMATION

The inside-outside algorithm (Baker, 1979) is a reestimation procedure for the rule probabilities of a Chomsky normal-form (CNF) SCFG. It takes as inputs an initial CNF SCFG and a training corpus of sentences and it iteratively reestimates rule probabilities to maximize the probability that the grammar used as a stochastic generator would produce the corpus.

A reestimation algorithm can be used both to refine the parameter estimates for a CNF SCFG derived by other means (Fujisaki et al., 1989) or to infer a grammar from scratch. In the latter case, the initial grammar for the inside-outside algorithm consists of all possible CNF rules over given sets N of nonterminals and Σ of terminals, with suitably assigned nonzero probabilities. In what follows, we will take N, Σ as fixed, $n = |N|$, $t = |\Sigma|$, and assume enumerations $N = \{A_1, \ldots, A_n\}$ and $\Sigma = \{b_1, \ldots, b_t\}$, with A_1 the grammar start symbol. A CNF SCFG over N, Σ can then be specified by the $n^3 + nt$ probabilities $B_{p,q,r}$ of each possible binary rule $A_p \rightarrow A_q\ A_r$ and $U_{p,m}$ of each possible unary rule $A_p \rightarrow b_m$. Since for each p the parameters $B_{p,q,r}$ and $U_{p,m}$ are supposed to be the probabilities of different ways of expanding A_p, we must have for all $1 \leq p \leq n$

$$\sum_{q,r} B_{p,q,r} + \sum_{m} U_{p,m} = 1 \qquad (7)$$

For grammar inference, we give random initial values to the parameters $B_{p,q,r}$ and $U_{p,m}$ subject to the constraints (7).

The intended meaning of rule probabilities in a SCFG is directly tied to the intuition of context-freeness: a derivation is assigned a probability which is the product of the probabilities of the rules used in each step of the derivation. Context-freeness together with the commutativity of multiplication thus allow us to identify all derivations associated to the same parse tree, and we will

$$I_p^c(i-1,i) = U_{p,m} \text{ where } c=(w,\mathcal{B}) \text{ and } b_m = w_i \tag{1}$$

$$I_p^c(i,k) = \bar{c}(i,k)\sum_{q,r}\sum_{i<j<k} B_{p,q,r} I_q^c(i,j) I_r^c(j,k) \tag{2}$$

$$O_p^c(0,|c|) = \begin{cases} 1 & \text{if } p=1 \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

$$O_p^c(i,k) = \bar{c}(i,k)\sum_{q,r}\left(\sum_{j=0}^{i-1} O_q^c(j,k) I_r^c(j,i) B_{q,r,p} + \sum_{j=k+1}^{|c|} O_q^c(i,j) B_{q,p,r} I_r^c(k,j)\right) \tag{4}$$

$$\hat{B}_{p,q,r} = \frac{\sum_{c\in C}\frac{1}{P^c}\sum_{0\le i<j<k\le|w|} B_{p,q,r} I_q^c(i,j) I_r^c(j,k) O_p^c(i,k)}{\sum_{c\in C} P_p^c/P^c} \tag{5}$$

$$\hat{U}_{p,m} = \frac{\sum_{c\in C}\frac{1}{P^c}\sum_{1\le i\le|c|, c=(w,\mathcal{B}), w_i=b_m} U_{p,m} O_p^c(i-1,i)}{\sum_{c\in C} P_p^c/P^c} \tag{6}$$

$$P^c = I_1^c(0,|c|)$$

$$P_p^c = \sum_{0\le i<j\le|c|} I_p^c(i,j) O_p^c(i,j)$$

Table 1: Bracketed Reestimation

speak indifferently below of derivation and analysis (parse tree) probabilities. Finally, the probability of a sentence or sentential form is the sum of the probabilities of all its analyses (equivalently, the sum of the probabilities of all of its leftmost derivations from the start symbol).

3.1. The Inside-Outside Algorithm

The basic idea of the inside-outside algorithm is to use the current rule probabilities and the training set W to estimate the expected frequencies of certain types of derivation step, and then compute new rule probability estimates as appropriate ratios of those expected frequency estimates. Since these are most conveniently expressed as relative frequencies, they are a bit loosely referred to as *inside* and *outside* probabilities. More precisely, for each $w \in W$, the inside probability $I_p^w(i,j)$ estimates the likelihood that A_p derives $_iw_j$, while the outside probability $O_p^w(i,j)$ estimates the likelihood of deriving sentential form $_0w_i\, A_p\, _jw$ from the start symbol A_1.

3.2. The Extended Algorithm

In adapting the inside-outside algorithm to partially bracketed training text, we must take into account the constraints that the bracketing imposes on possible derivations, and thus on possible phrases. Clearly, nonzero values for $I_p^w(i,j)$ or $O_p^w(i,j)$ should only be allowed if $_iw_j$ is compatible with the bracketing of w, or, equivalently, if (i,j) is valid for the bracketing of w. Therefore, we will in the following assume a corpus C of bracketed strings $c=(w,\mathcal{B})$, and will modify the standard formulas for the inside and outside probabilities and rule probability reestimation (Baker, 1979; Lari and Young, 1990; Jelinek et al., 1990) to involve only constituents whose spans are compatible with string bracketings. For this purpose, for each bracketed string $c=(w,\mathcal{B})$ we define the auxiliary function

$$\bar{c}(i,j) = \begin{cases} 1 & \text{if } (i,j) \text{ is valid for } b\in\mathcal{B} \\ 0 & \text{otherwise} \end{cases}$$

The reestimation formulas for the extended algorithm are shown in Table 1. For each bracketed sentence c in the training corpus, the inside probabilities of longer spans of c are computed from

those for shorter spans with the recurrence given by equations (1) and (2). Equation (2) calculates the expected relative frequency of derivations of ${}_iw_k$ from A_p compatible with the bracketing $\mathcal{B}$ of $c = (w, \mathcal{B})$. The multiplier $\bar{c}(i,k)$ is 1 just in case (i,k) is valid for $\mathcal{B}$, that is, when A_p can derive ${}_iw_k$ compatibly with $\mathcal{B}$.

Similarly, the outside probabilities for shorter spans of c can be computed from the inside probabilities and the outside probabilities for longer spans with the recurrence given by equations (3) and (4). Once the inside and outside probabilities computed for each sentence in the corpus, the reestimated probability of binary rules, $\hat{B}_{p,q,r}$, and the reestimated probability of unary rules, $\hat{U}_{p,m}$, are computed by the reestimation formulas (5) and (6), which are just like the original ones (Baker, 1979; Jelinek et al., 1990; Lari and Young, 1990) except for the use of bracketed strings instead of unbracketed ones.

The denominator of ratios (5) and (6) estimates the probability that a compatible derivation of a bracketed string in C will involve at least one expansion of nonterminal A_p. The numerator of (5) estimates the probability that a compatible derivation of a bracketed string in C will involve rule $A_p \rightarrow A_q\ A_r$, while the numerator of (6) estimates the probability that a compatible derivation of a string in C will rewrite A_p to b_m. Thus (5) estimates the probability that a rewrite of A_p in a compatible derivation of a bracketed string in C will use rule $A_p \rightarrow A_q\ A_r$, and (6) estimates the probability that an occurrence of A_p in a compatible derivation of a string in in C will be rewritten to b_m. These are the best current estimates for the binary and unary rule probabilities.

The process is then repeated with the reestimated probabilities until the increase in the estimated probability of the training text given the model becomes negligible, or, what amounts to the same, the decrease in the cross entropy estimate (negative log probability)

$$\hat{H}(C,G) = -\frac{\sum_{c \in C} \log P^c}{\sum_{c \in C} |c|} \qquad (8)$$

becomes negligible. Note that for comparisons with the original algorithm, we should use the cross-entropy estimate $\hat{H}(W,G)$ of the *unbracketed* text W with respect to the grammar G, not (8).

3.3. Complexity

Each of the three steps of an iteration of the original inside-outside algorithm — computation of inside probabilities, computation of outside probabilities and rule probability reestimation – takes time $O(|w|^3)$ for each training sentence w. Thus, the whole algorithm is $O(|w|^3)$ on each training sentence.

However, the extended algorithm performs better when bracketing information is provided, because it does not need to consider all possible spans for constituents, but only those compatible with the training set bracketing. In the limit, when the bracketing of each training sentence comes from a complete binary-branching analysis of the sentence (a *full* binary bracketing), the time of each step reduces to $O(|w|)$. This can be seen from the following three facts about any full binary bracketing $\mathcal{B}$ of a string w:

1. $\mathcal{B}$ has $O(|w|)$ spans;
2. For each (i,k) in $\mathcal{B}$ there is exactly one *split point* j such that both (i,j) and (j,k) are in $\mathcal{B}$;
3. Each valid span with respect to $\mathcal{B}$ must already be a member of $\mathcal{B}$.

Thus, in equation (2) for instance, the number of spans (i,k) for which $\bar{c}(i,k) \neq 0$ is $O(|c|)$, and there is a single j between i and k for which $\bar{c}(i,j) \neq 0$ and $\bar{c}(j,k) \neq 0$. Therefore, the total time to compute all the $I_p^c(i,k)$ is $O(|c|)$. A similar argument applies to equations (4) and (5).

Note that to achieve the above bound as well as to take advantage of whatever bracketing is available to improve performance, the implementation must preprocess the training set appropriately so that the valid spans and their split points are efficiently enumerated.

4. EXPERIMENTAL EVALUATION

The following experiments, although preliminary, give some support to our earlier suggested advantages of the inside-outside algorithm for partially bracketed corpora.

The first experiment involves an artificial example used by Lari and Young (1990) in a previous evaluation of the inside-outside algorithm. In this

case, training on a bracketed corpus can lead to a good solution while no reasonable solution is found training on raw text only.

The second experiment uses a naturally occurring corpus and its partially bracketed version provided by the Penn Treebank (Brill et al., 1990). We compare the bracketings assigned by grammars inferred from raw and from bracketed training material with the Penn Treebank bracketings of a separate test set.

To evaluate objectively the accuracy of the analyses yielded by a grammar G, we use a Viterbi-style parser to find the most likely analysis of each test sentence according to G, and define the *bracketing accuracy* of the grammar as the proportion of phrases in those analyses that are compatible in the sense defined in Section 2 with the tree bank bracketings of the test set. This criterion is closely related to the "crossing parentheses" score of Black *et al.* (1991).[1]

In describing the experiments, we use the notation G_{R} for the grammar estimated by the original inside-outside algorithm, and G_{B} for the grammar estimated by the bracketed algorithm.

4.1. Inferring the Palindrome Language

We consider first an artificial language discussed by Lari and Young (1990). Our training corpus consists of 100 sentences in the palindrome language L over two symbols a and b

$$L = \{w\, w^R \mid w \in \{a,b\}^*\}.$$

randomly generated with the SCFG

$$\begin{array}{lcl} S & \xrightarrow{0.4} & A\,C \\ S & \xrightarrow{0.4} & B\,D \\ S & \xrightarrow{0.1} & A\,A \\ S & \xrightarrow{0.1} & B\,B \\ C & \xrightarrow{1} & S\,A \\ D & \xrightarrow{1} & S\,B \\ A & \xrightarrow{1} & a \\ B & \xrightarrow{1} & b \end{array}$$

[1] Since the grammar inference procedure is restricted to Chomsky normal form grammars, it cannot avoid difficult decisions by leaving out brackets (thus making flatter parse trees), as human annotators often do. Therefore, the recall component in Black *et al.*'s figure of merit for parser is not needed.

The initial grammar consists of all possible CNF rules over five nonterminals and the terminals a and b (135 rules), with random rule probabilities.

As shown in Figure 1, with an unbracketed training set W the cross-entropy estimate $\hat{H}(W, G_{\mathrm{R}})$ remains almost unchanged after 40 iterations (from 1.57 to 1.44) and no useful solution is found. In contrast, with a fully bracketed version C of the same training set, the cross-entropy estimate $\hat{H}(W, G_{\mathrm{B}})$ decreases rapidly (1.57 initially, 0.88 after 21 iterations). Similarly, the cross-entropy estimate $\hat{H}(C, G_{\mathrm{B}})$ of the bracketed text with respect to the grammar improves rapidly (2.85 initially, 0.89 after 21 iterations).

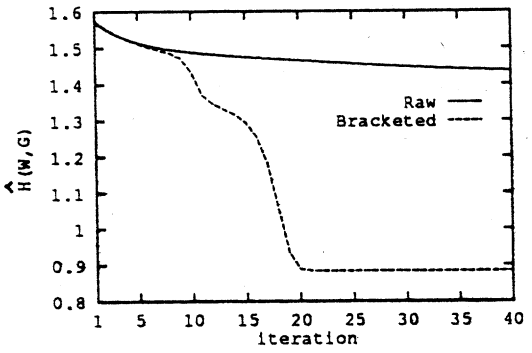

Figure 1: Convergence for the Palindrome Corpus

The inferred grammar models correctly the palindrome language. Its high probability rules ($p > 0.1$, $p/p' > 30$ for any excluded rule p') are

$$\begin{array}{lcl} S & \rightarrow & A\,D \\ S & \rightarrow & C\,B \\ B & \rightarrow & S\,C \\ D & \rightarrow & S\,A \\ A & \rightarrow & b \\ B & \rightarrow & a \\ C & \rightarrow & a \\ D & \rightarrow & b \end{array}$$

which is a close to optimal CNF CFG for the palindrome language.

The results on this grammar are quite sensitive to the size and statistics of the training corpus and the initial rule probability assignment. In fact, for a couple of choices of initial grammar and corpus, the original algorithm produces grammars with somewhat better cross-entropy estimates than those yielded by the new one. However, in every case the bracketing accuracy on

a separate test set for the result of bracketed training is above 90% (100% in several cases), in contrast to bracketing accuracies ranging between 15% and 69% for unbracketed training.

4.2. Experiments on the ATIS Corpus

For our main experiment, we used part-of-speech sequences of spoken-language transcriptions in the Texas Instruments subset of the Air Travel Information System (ATIS) corpus (Hemphill et al., 1990), and a bracketing of those sequences derived from the parse trees for that subset in the Penn Treebank.

Out of the 770 bracketed sentences (7812 words) in the corpus, we used 700 as a training set C and 70 (901 words) as a test set T. The following is an example training string

```
( ( ( VB ( DT NNS ( IN ( ( NN ) (
  NN CD ) ) ) ) ) ) . )
```

corresponding to the parsed sentence

```
(((List (the fares (for ((flight)
  (number 891)))))) .)
```

The initial grammar consists of all 4095 possible CNF rules over 15 nonterminals (the same number as in the tree bank) and 48 terminal symbols for part-of-speech tags.

A random initial grammar was trained separately on the unbracketed and bracketed versions of the training corpus, yielding grammars G_R and G_B.

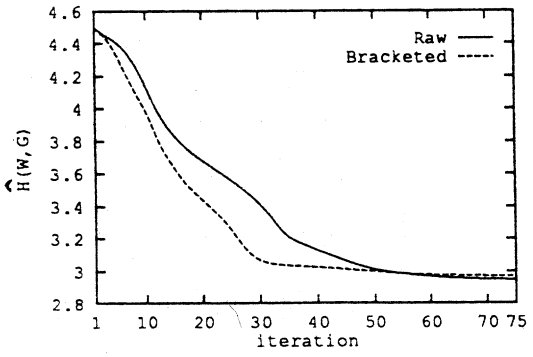

Figure 2: Convergence for the ATIS Corpus

Figure 2 shows that $\hat{H}(W, G_B)$ initially decreases faster than the $\hat{H}(W, G_R)$, although eventually the two stabilize at very close values: after 75 iterations, $\hat{H}(W, G_B) \approx 2.97$ and $\hat{H}(W, G_R) \approx 2.95$. However, the analyses assigned by the resulting grammars to the test set are drastically different.

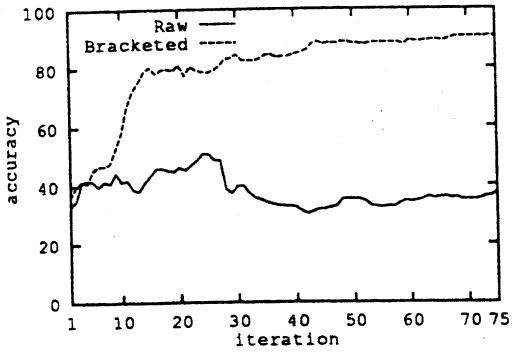

Figure 3: Bracketing Accuracy for the ATIS Corpus

With the training and test data described above, the bracketing accuracy of G_R after 75 iterations was only 37.35%, in contrast to 90.36% bracketing accuracy for G_B. Plotting bracketing accuracy against iterations (Figure 3), we see that unbracketed training does not on the whole improve accuracy. On the other hand, bracketed training steadily improves accuracy, although not monotonically.

It is also interesting to look at some the differences between G_R and G_B, as seen from the most likely analyses they assign to certain sentences. Table 2 shows two bracketed test sentences followed by their most likely G_R and G_B analyses, given for readability in terms of the original words rather than part-of-speech tags.

For test sentence (A), the only G_B constituent not compatible with the tree bank bracketing is **(Delta flight number)**, although the constituent **(the cheapest)** is linguistically wrong. The appearance of this constituent can be explained by lack of information in the tree bank about the internal structure of noun phrases, as exemplified by tree bank bracketing of the same sentence. In contrast, the G_R analysis of the same string contains 16 constituents incompatible with the tree bank.

For test sentence (B), the G_B analysis is fully compatible with the tree bank. However, the G_R analysis has nine incompatible constituents, which for

(A)	(I would (like (to (take (Delta ((flight number) 83)) (to Atlanta)))).) (What ((is (the cheapest fare (I can get)))) ?)
G_R	(I (would (like ((to ((take (Delta flight)) (number (83 ((to Atlanta) .))))) ((What (((is the) cheapest) fare)) ((I can) (get ?)))))))
G_B	(((I (would (like (to (take (((Delta (flight number)) 83) (to Atlanta))))))) .) ((What (is (((the cheapest) fare) (I (can get))))) ?))
(B)	((Tell me (about (the public transportation ((from SFO) (to San Francisco))))).)
G_R	(Tell ((me (((about the) public) transportation)) ((from SFO) ((to San) (Francisco .)))))
G_B	((Tell (me (about (((the public) transportation) ((from SFO) (to (San Francisco))))))) .)

Table 2: Comparing Bracketings

example places **Francisco** and the final punctuation in a lowest-level constituent. Since final punctuation is quite often preceded by a noun, a grammar inferred from raw text will tend to bracket the noun with the punctuation mark.

This experiment illustrates the fact that although SCFGs provide a hierarchical model of the language, that structure is undetermined by raw text and only by chance will the inferred grammar agree with qualitative linguistic judgments of sentence structure. This problem has also been previously observed with linguistic structure inference methods based on mutual information. Magerman and Marcus (1990) addressed the problem by specifying a predetermined list of pairs of parts of speech (such as verb-preposition, pronoun-verb) that can never be embraced by a low-level constituent. However, these constraints are stipulated in advance rather than being automatically derived from the training material, in contrast with what we have shown to be possible with the inside-outside algorithm for partially bracketed corpora.

5. CONCLUSIONS AND FURTHER WORK

We have introduced a modification of the well-known inside-outside algorithm for inferring the parameters of a stochastic context-free grammar that can take advantage of constituent information (constituent bracketing) in a partially bracketed corpus.

The method has been successfully applied to SCFG inference for formal languages and for part-of-speech sequences derived from the ATIS spoken-language corpus.

The use of partially bracketed corpus can reduce the number of iterations required for convergence of parameter reestimation. In some cases, a good solution is found from a bracketed corpus but not from raw text. Most importantly, the use of partially bracketed natural corpus enables the algorithm to infer grammars specifying linguistically reasonable constituent boundaries that cannot be inferred by the inside-outside algorithm on raw text. While none of this is very surprising, it supplies some support for the view that purely unsupervised, self-organizing grammar inference methods may have difficulty in distinguishing between underlying grammatical structure and contingent distributional regularities, or, to put it in another way, it gives some evidence for the importance of nondistributional regularities in language, which in the case of bracketed training have been supplied indirectly by the linguists carrying out the bracketing.

Also of practical importance, the new algorithm can have better time complexity for bracketed text. In the best situation, that of a training set with full binary-branching bracketing, the time for each iteration is in fact linear on the total length of the set.

These preliminary investigations could be extended in several ways. First, it is important to determine the sensitivity of the training algorithm to the initial probability assignments and training corpus, as well as to lack or misplacement of brackets. We have started experiments in this direction, but reasonable statistical models of bracket elision

and misplacement are lacking.

Second, we would like to extend our experiments to larger terminal vocabularies. As is well known, this raises both computational and data sparseness problems, so clustering of terminal symbols will be essential.

Finally, this work does not address a central weakness of SCFGs, their inability to represent lexical influences on distribution except by a statistically and computationally impractical proliferation of nonterminal symbols. One might instead look into versions of the current algorithm for more lexically-oriented formalisms such as stochastic lexicalized tree-adjoining grammars (Schabes, 1992).

ACKNOWLEGMENTS

We thank Aravind Joshi and Stuart Shieber for useful discussions, and Mitch Marcus, Beatrice Santorini and Mary Ann Marcinkiewicz for making available the ATIS corpus in the Penn Treebank. The second author is partially supported by DARPA Grant N0014-90-31863, ARO Grant DAAL03-89-C-0031 and NSF Grant IRI90-16592.

REFERENCES

J.K. Baker. 1979. Trainable grammars for speech recognition. In Jared J. Wolf and Dennis H. Klatt, editors, *Speech communication papers presented at the 97th Meeting of the Acoustical Society of America*, MIT, Cambridge, MA, June.

E. Black, S. Abney, D. Flickenger, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of english grammars. In *DARPA Speech and Natural Language Workshop*, pages 306–311, Pacific Grove, California. Morgan Kaufmann.

T. Booth. 1969. Probabilistic representation of formal languages. In *Tenth Annual IEEE Symposium on Switching and Automata Theory*, October.

Eric Brill, David Magerman, Mitchell Marcus, and Beatrice Santorini. 1990. Deducing linguistic structure from the statistics of large corpora. In *DARPA Speech and Natural Language Workshop*. Morgan Kaufmann, Hidden Valley, Pennsylvania, June.

T. Fujisaki, F. Jelinek, J. Cocke, E. Black, and T. Nishino. 1989. A probabilistic parsing method for sentence disambiguation. In *Proceedings of the International Workshop on Parsing Technologies*, Pittsburgh, August.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *DARPA Speech and Natural Language Workshop*, Hidden Valley, Pennsylvania, June.

F. Jelinek, J. D. Lafferty, and R. L. Mercer. 1990. Basic methods of probabilistic context free grammars. Technical Report RC 16374 (72684), IBM, Yorktown Heights, New York 10598.

Frederick Jelinek, Robert L. Mercer, and Salim Roukos. 1992. Principles of lexical language modeling for speech recognition. In Sadaoki Furui and M. Mohan Sondhi, editors, *Advances in Speech Signal Processing*, pages 651–699. Marcel Dekker, Inc., New York, New York.

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.

K. Lari and S. J. Young. 1991. Applications of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech and Language*, 5:237–257.

David Magerman and Mitchell Marcus. 1990. Parsing a natural language using mutual information statistics. In *AAAI-90*, Boston, MA.

Yves Schabes. 1992. Stochastic lexicalized tree-adjoining grammars. In *COLING 92*. Forthcoming.