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

Yves Schabes

Dept. of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104-6389

ABSTRACT

The inside-outside algorithm for inferring the parameters of a stochastic context-free grammar is extended to take advantage of constituent information in a partially parsed corpus. Experiments on formal and natural language parsed corpora show that the new algorithm can achieve faster convergence and better modelling of hierarchical structure than the original one. In particular, over 90% of the constituents in the most likely analyses of a test set are compatible with test set constituents for a grammar trained on a corpus of 700 hand-parsed part-of-speech strings for ATIS sentences.

1. MOTIVATION

Grammar inference is a challenging problem for statistical approaches to natural-language processing. The most successful grammar inference techniques involve stochastic finite-state language models such as hidden Markov models (HMMs) [1]. However, finite-state language models fail to represent the hierarchical structure of natural language. Therefore, stochastic versions of grammar formalisms structurally more expressive are worth investigating. Baker [2] generalized the parameter estimation methods for HMMs to stochastic context-free grammars (SCFGs) [3] as the inside-outside algorithm. Unfortunately, the application of SCFGs and the inside-outside algorithm to natural-language modeling [4, 5, 6] has so far been inconclusive.

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. 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, 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. For example, since in English texts pronouns are very likely to immediately precede a verb, a grammar inferred from raw text will tend to together the subject pronoun with the 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 important advantages. We empirically show that the use of partially parsed corpus can decrease the number of iterations needed to reach a solution. We also exhibit cases where a good solution is found from partially parsed corpus but not from raw text. Most importantly, the use of partially parsed corpus enables the algorithm to infer grammars that derive constituent boundaries that cannot be inferred from raw text.

We first outline our extension of the inside-outside algorithm to partially parsed text, and then report preliminary experiments illustrating the advantages of the extended algorithm.

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 $0 \leq i < j \leq |w|$. By convention, span (i, j) delimits substring ${}_iw_j = w_{i+1} \cdots w_j$ of w. We also

use the abbreviation ${}_i w$ for ${}_i w_{|w|}$.

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 (sequence of) constituents ${}_i w_j$. 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 $k < i < l < j$. We also say that two bracketings of the same string are *compatible* if their union is consistent.

Note that there is no requirement that a bracketing of w describe fully the 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 production 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 no span of a symbol occurrence in the derivation overlaps a span in $\mathcal{B}$.

3. THE INSIDE-OUTSIDE ALGORITHM

The inside-outside algorithm [2] 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 [7] 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 suitable 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 the for all $1 \leq p \leq n$

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

For grammar inference, we give random initial values to the parameters $B_{p,q,r}$ and $U_{p,m}$ subject to the constraints (1).

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 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).

The basic idea of the inside-outside algorithm is to use the current rule probabilities to estimate from the training text the expected frequencies of certain derivation steps, and then compute new rule probability estimates as appropriate frequency ratios. Therefore, each iteration of the algorithm starts by calculating estimates of the number of occurrences of the relevant configurations in each of the sentences w in the training corpus W. Because the frequency estimates are most conveniently computed as ratios of other frequencies, they are a bit loosely referred to as *inside* and *outside* probabilities.

In the original inside-outside algorithm, for each $w \in W$, the inside probability $I_p^w(i, j)$ estimates the likelihood that A_p derives ${}_i w_j$, while the outside probability $O_p^w(i, j)$ estimates the likelihood of deriving sentential form ${}_0 w_i\ A_p\ {}_j w$ from the start symbol A_1. In adapting the algorithm to partially bracketed strings 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 ${}_i w_j$ is compatible with the bracketing of w, or, equivalently, if (i, j) does not overlap any span

in the bracketing of w. Therefore, we will in the following assume a bracketed corpus C, which as described above is a set of bracketed strings $c = (w, \mathcal{B})$, and will modify the standard formulae for the inside and outside probabilities and rule probability reestimation [2, 4, 5] 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

$$\bar{c}(i,j) = \begin{cases} 1 & \text{if } (i,j) \text{ does not overlap any } b \in \mathcal{B} \\ 0 & \text{otherwise} \end{cases}$$

For each bracketed sentence c in the training corpus, the inside probabilities of longer spans of c can be computed from those for shorter spans by the following recurrence equations:

$$I_p^c(i-1,i) = U_{p,m} \text{ where } c = (w, \mathcal{B}) \text{ and } b_m = w_i \quad (2)$$

$$I_p^c(i,k) = \bar{c}(i,k) \sum_{q,r} \sum_{i<j<k} B_{p,q,r} I_q^c(i,j) I_r^c(j,k) \quad (3)$$

Equation (3) computes 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 0 just in case (i,k) overlaps some span in $\mathcal{B}$, which is exactly when A_p cannot 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 by the following recurrence:

$$O_p^c(0,|c|) = \begin{cases} 1 & \text{if } p = 1 \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

$$O_p^c(i,k) = \bar{c}(i,k) \sum_{q,r} \left(\begin{array}{c} \sum_{j=0}^{i-1} O_q^c(j,k) I_r^c(j,i) B_{q,r,p} \\ + \\ \sum_{j=k+1}^{|c|} O_q^c(i,j) B_{q,p,r} I_r^c(k,j) \end{array} \right) \quad (5)$$

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 using the following reestimation formulae, which are just like the standard ones [2, 5, 4] except for the use of bracketed strings instead of unbracketed ones:

$$\hat{B}_{p,q,r} = \frac{\sum_{c \in C} \frac{1}{P^c} \sum_{0 \le i < j < k \le |w|} \left(\begin{array}{c} B_{p,q,r} I_q^c(i,j) \\ \times \\ I_r^c(j,k) O_p^c(i,k) \end{array} \right)}{\sum_{c \in C} P_p^c / P^c} \quad (6)$$

$$\hat{U}_{p,m} = \frac{\sum_{c \in C} \frac{1}{P^c} \sum_{1 \le i \le |c|, c=(w,\mathcal{B}), w_i = b_m} U_{p,m} O_p^c(i-1,i)}{\sum_{c \in C} P_p^c / P^c} \quad (7)$$

where P^c is the probability assigned by the current model to bracketed string c

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

and P_p^c is the probability assigned by the current model to the set of derivations compatible with c involving some instance of nonterminal A_p

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

The denominator of ratios (6) and (7) 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 (6) 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 (7) estimates the probability that a compatible derivation of a string in C will rewrite A_p to b_m. Thus (6) 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 (7) 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. Clearly, 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 (log probability)

$$H_e(C) = -\frac{\sum_{c \in C} \log Prob(c)}{\sum_{c \in C} |c|} = -\frac{\sum_{c \in C} \log I_1^c(0,|c|)}{\sum_{c \in C} |c|} \quad (8)$$

becomes negligible. Note that for comparisons with the original algorithm, we should use the cross entropy of the *unbracketed* text with respect to the grammar, not (8).

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.

We start with a formal-language example used by Lari and Young [4] 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.

Then, using a naturally occurring corpus and its partially bracketed version provided by the Penn Treebank, we compare the bracketings assigned by grammars inferred from raw and from bracketed training material with the Penn Treebank bracketings.

Together, the experiments support the view that training on bracketed corpora can lead to better convergence, and the resulting grammars agree better with linguistic judgments of sentence structure.

4.1. Inferring the Palindrome Language

We consider first an artificial language discussed by Lari and Young [4]. 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

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

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

As shown in Figure 1, with an unbracketed training set the log probability remains almost unchanged after 40 iterations (from 1.57 to 1.43) and no useful solution is found. In contrast, with the same training set fully bracketed, the log probability of the inferred grammar computed on the raw text decreases rapidly (1.57 initially, 0.87 after 22 iterations). Similarly, the cross entropy estimate of the bracketed text with respect to the grammar improves rapidly (2.85 initially, 0.87 after 22 iterations).

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

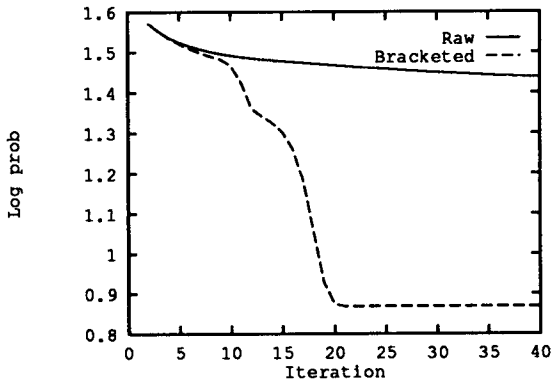

Figure 1: Convergence for the Palindrome Corpus

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

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 yields somewhat better results than the new one. However, in no experiment did the training on unparsed text achieve nearly as good a result as that shown above for parsed text.

4.2. Experiments on the ATIS Corpus

We also conducted an experiment on inferring grammars for the language consisting of part-of-speech sequences of spoken-language transcriptions in the Texas Instruments subset of the Air Travel Information System (ATIS) corpus [8]. We take advantage of the availability of the hand-parsed version of the ATIS corpus provided by the Penn Treebank project [9] and use the corresponding bracketed corpus over parts of speech as training data.

Out of the 770 bracketed sentences (7812 words) in the corpus, we used 700 as training data and 70 (901 words) as test set. 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 possible CNF rules (4095 rules) over 15 nonterminals (the same number as in the tree bank) and 48 terminals corresponding to the parts of speech used in the tree bank.

We trained a random initial grammar twice, on the unbracketed version of the training corpus yielding grammar G_R, and on the bracketed training set, yielding grammar G_B.

Figure 2 shows that the convergence to G_B is faster than the convergence to G_R. Even though the cross-entropy estimates for the raw training text with both grammars are not that different after 50 iterations (3.0 for G_B, 3.02 for G_R), the analyses assigned by the resulting grammars to the test set are drastically different.

To evaluate objectively the quality of the analyses yielded by a grammar G, we used a Viterbi-style parser to find the most likely analyses of the test set according to G, and computed 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.* [10]. We found that that only 35% of the constituents in the most likely G_R analyses of the test set are compatible with tree bank bracketing, in contrast to 88% of the constituents in the most likely G_B analysis.

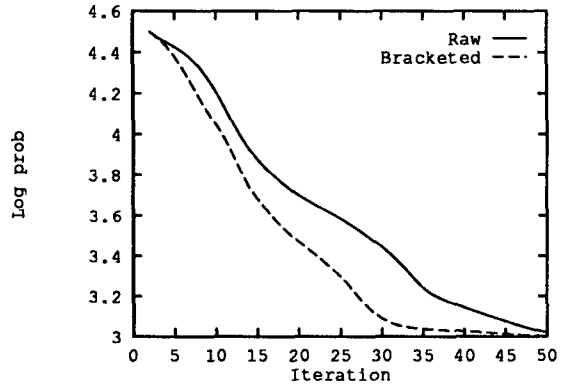

Figure 2: Convergence for the ATIS Corpus

It is 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. For readability, we give the analyses in terms of the original words rather than part of speech tags.

As a first example, G_B gives the following bracketings:

```
((((I (would (like (to (take (((Delta
   (flight number)) 83) (to
   Atlanta))))))) .)
((What (is (((the cheapest) fare) (I
   (can get))))) ?))
```

Although the constituent **(the cheapest)** is linguistically wrong, the only constituent not compatible with the tree bank bracketing is **(Delta flight number)**:

```
(I would (like (to (take (Delta
   ((flight number) 83)) (to
   Atlanta)))).)
(What ((is (the cheapest fare (I can
   get)))) ?)
```

In contrast, G_R gives the following analyses for the same strings, with 16 constituents incompatible with the tree bank:

```
(I (would (like ((to ((take (Delta
   flight)) (number (83 ((to Atlanta)
   .)))))
((What (((is the) cheapest) fare)) ((I
   can) (get ?)))))))
```

Another example analysis for G_B is

```
((Tell (me (about (((the public)
   transportation) ((from SFO) (to
   (San Francisco))))))) .)
```

which is compatible with the tree bank one:

```
((Tell me (about (the public
   transportation ((from SFO) (to San
   Francisco))))).)
```

However, the most likely G_R analysis has nine constituents incompatible with the tree bank:

```
(Tell ((me (((about the) public)
   transportation)) ((from SFO) ((to
   San) (Francisco .)))))
```

In this analysis, a **Francisco** and the final punctuation are places 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.

Even better results can be obtained by continuing the reestimation on bracketed text. After 78 iterations, 91% of the constituents of the most likely parse of the test set are compatible with the tree bank bracketing.

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 [11] propose to alleviate this behavior by enforcing that a predetermined list of pairs of words (such as verb-preposition, pronoun-verb) are never embraced by a 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.

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 [12].

ACKNOWLEGMENTS

We thank Aravind Joshi and Stuart Shieber for useful discussions. The second author is partially supported by DARPA Grant N0014-90-31863, ARO Grant DAAL03-89-C-0031 and NSF Grant IRI90-16592.

REFERENCES

1. Rabiner, L. R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, February 1989.
2. Baker, J. Trainable grammars for speech recognition. In Wolf, J. J. and Klatt, D. H., editors, *Speech communication papers presented at the 97th Meeting of the Acoustical Society of America*, MIT, Cambridge, MA, June 1979.
3. Booth, T. Probabilistic representation of formal languages. In *Tenth Annual IEEE Symposium on Switching and Automata Theory*, October 1969.
4. Lari, K. and Young, S. J. The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech and Language*, 4:35–56, 1990.
5. Jelinek, F., Lafferty, J. D., and Mercer, R. L. Basic methods of probabilistic context free grammars. Technical Report RC 16374 (72684), IBM, Yorktown Heights, New York 10598, 1990.
6. Lari, K. and Young, S. J. Applications of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech and Language*, 5:237–257, 1991.
7. Fujisaki, T., Jelinek, F., Cocke, J., Black, E., and Nishino, T. A probabilistic parsing method for sentence disambiguation. In *Proceedings of the International Workshop on Parsing Technologies*, Pittsburgh, August 1989.
8. Hemphill, C. T., Godfrey, J. J., and Doddington, G. R. The ATIS spoken language systems pilot corpus. In *DARPA Speech and Natural Language Workshop*, Hidden Valley, Pennsylvania, June 1990.
9. Brill, E., Magerman, D., Marcus, M., and Santorini, B. Deducing linguistic structure from the statistics of large corpora. In *DARPA Speech and Natural Language Workshop*. Morgan Kaufmann, Hidden Valley, Pennsylvania, June 1990.
10. Black, E., Abney, S., Flickenger, D., Grishman, R., harrison, P., Hindle, D., Ingria, R., Jelinek, F., Klavans, J., Liberman, M., Marcus, M., Roukos, S., Santorini, B., and Strzalkowski, T. A procedure for quantitatively comparing the syntactic coverage of english grammars. In *DARPA Speech and Natural Language Workshop*, pages 306–311, Pacific Grove, California, 1991. Morgan Kaufmann.
11. Magerman, D. and Marcus, M. Parsing a natural language using mutual information statistics. In *AAAI-90*, Boston, MA, 1990.
12. Schabes, Y. Stochastic lexicalized tree-adjoining grammars. Also in these proceedings, 1992.