# Using an Annotated Language Corpus as a Virtual Stochastic Grammar

**Rens Bod**

Department of Computational Linguistics
University of Amsterdam
Spuistraat 134
NL-1012 VB Amsterdam
rens@alf.let.uva.nl

**Abstract**

In Data Oriented Parsing (DOP), an annotated language corpus is used as a virtual stochastic grammar. An input string is parsed by combining subtrees from the corpus. As a consequence, one parse tree can usually be generated by several derivations that involve different subtrees. This leads to a statistics where the probability of a parse is equal to the sum of the probabilities of all its derivations. In (Scha, 1990) an informal introduction to DOP is given, while (Bod, 1992) provides a formalization of the theory. In this paper we show that the maximum probability parse can be estimated in polynomial time by applying Monte Carlo techniques. The model was tested on a set of hand-parsed strings from the Air Travel Information System (ATIS) corpus. Preliminary experiments yield 96% test set parsing accuracy.

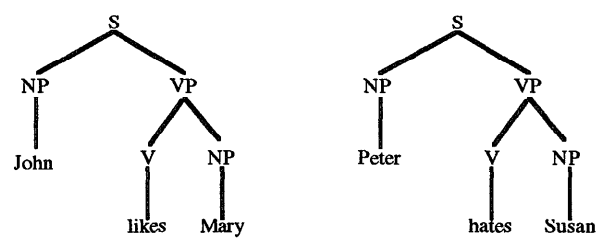
## Motivation

As soon as a formal grammar characterizes a non-trivial part of a natural language, almost every input string of reasonable length gets an unmanageably large number of different analyses. Since most of these analyses are not perceived as plausible by a human language user, there is a need for distinguishing the plausible parse(s) of an input string from the implausible ones. In stochastic language processing, it is assumed that the most plausible parse of an input string is its most probable parse. Most instantiations of this idea estimate the probability of a parse by assigning application probabilities to context free rewrite rules (Jelinek, 1990), or by assigning combination probabilities to elementary structures (Resnik, 1992; Schabes, 1992).

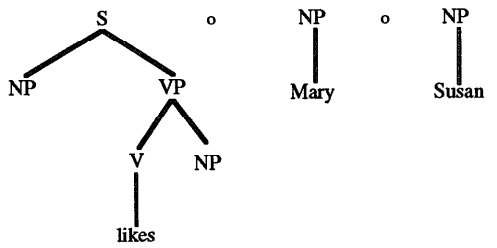
There is some agreement now that context free rewrite rules are not adequate for estimating the probability of a parse, since they cannot capture syntactic/lexical context, and hence cannot describe how the probability of syntactic structures or lexical items depends on that context. In stochastic tree-adjoining grammar (Schabes, 1992), this lack of context-sensitivity is overcome by assigning probabilities to larger structural units. However, it is not always evident which structures should be considered as elementary structures. In (Schabes, 1992) it is proposed to infer a stochastic TAG from a large training corpus using an inside-outside-like iterative algorithm.

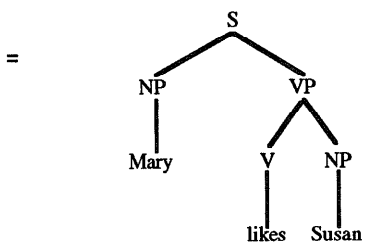
Data Oriented Parsing (DOP) (Scha, 1990; Bod, 1992), distinguishes itself from other statistical approaches in that it omits the step of inferring a grammar from a corpus. Instead, an annotated corpus is directly used as a stochastic grammar. An input string is parsed by combining subtrees from the corpus. In this view, every subtree can be considered as an elementary structure. As a consequence, one parse tree can usually be generated by several derivations that involve different subtrees. This leads to a statistics where the probability of a parse is equal to the sum of the probabilities of all its derivations. It is hoped that this approach can accommodate all statistical properties of a language corpus.

Let us illustrate DOP with an extremely simple example. Suppose that a corpus consists of only two trees:

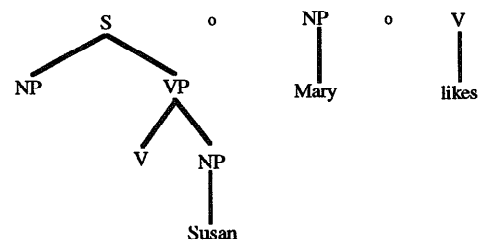



Suppose that our combination operation (indicated with ∘) consists of substituting a subtree on the leftmost identically labeled leaf node of another subtree. Then the sentence *Mary likes Susan* can be parsed as an *S* by combining the following subtrees from the corpus.

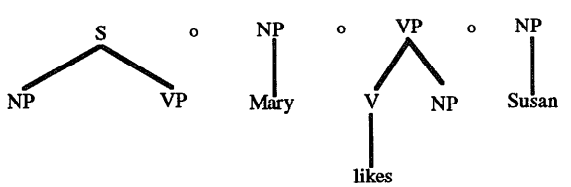



But the same parse tree can also be derived by combining other subtrees, for instance:




or




Thus, a parse can have several derivations involving different subtrees. These derivations have different probabilities. Using the corpus as our stochastic grammar, we estimate the probability of substituting a certain subtree on a specific node as the probability of selecting this subtree among all subtrees in the corpus that could be substituted on that node. The probability of a derivation can be computed as the product of the probabilities of the subtrees that are combined. For the example derivations above, this yields:

$$
\begin{aligned}
P(\text{1st example}) &= 1/20 * 1/4 * 1/4 &&= 1/320 \\
P(\text{2nd example}) &= 1/20 * 1/4 * 1/2 &&= 1/160 \\
P(\text{3rd example}) &= 2/20 * 1/4 * 1/8 * 1/4 &&= 1/1280
\end{aligned}
$$

This example illustrates that a statistical language model which defines probabilities over parses by taking into account only one derivation, does not accommodate all statistical properties of a language corpus. Instead, we will define the probability of a parse as the sum of the probabilities of all its derivations. Finally, the probability of a string is equal to the sum of the probabilities of all its parses.

We will show that conventional parsing techniques can be applied to DOP, but that this becomes very inefficient, since the number of derivations of a parse grows exponentially with the length of the input string. However, we will show that DOP can be parsed in polynomial time by using Monte Carlo techniques.

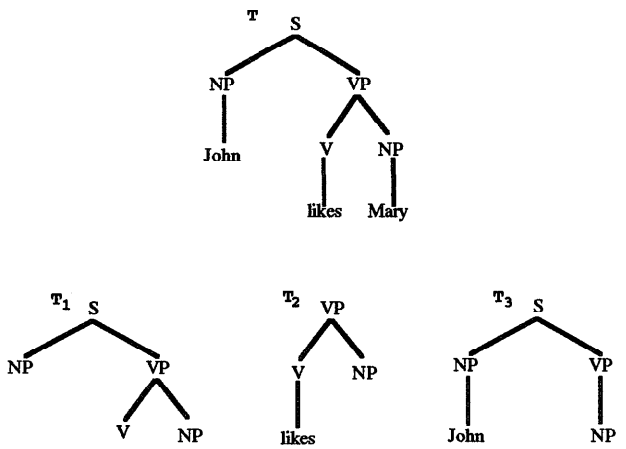
An important advantage of using a corpus for probability calculation, is that no training of parameters is needed, as is the case for other stochastic grammars (Jelinek et al., 1990; Pereira and Schabes, 1992; Schabes, 1992). Secondly, since we take into account all derivations of a parse, no relationship that might possibly be of statistical interest is ignored.

## The Model

As might be clear by now, a DOP-model is characterized by a corpus of tree structures, together with a set of operations that combine subtrees from the corpus into new trees. In this section we explain more precisely what we mean by subtree, operations etc., in order to arrive at definitions of a parse and the probability of a parse with respect to a corpus. For a treatment of DOP in more formal terms we refer to (Bod, 1992).

### Subtree

A *subtree* of a tree $T$ is a connected subgraph $S$ of $T$ such that for every node in $S$ holds that if it has daughter nodes, then these are equal to the daughter nodes of the corresponding node in $T$. It is trivial to see that a subtree is also a tree. In the following example $T_1$ and $T_2$ are subtrees of T, whereas $T_3$ isn't.




The general definition above also includes subtrees consisting of one node. Since such subtrees do not contribute to the parsing process, we exclude these pathological cases and consider as the set of subtrees the non-trivial ones consisting of more than one node. We shall use the following notation to indicate that a tree $t$ is a non-trivial subtree of a tree in a corpus $C$:

$$t \varepsilon C \;=_{def}\; \exists T \in C: t \text{ is a non-trivial subtree of } T$$

## Operations

In this article we will limit ourselves to the basic operation of *substitution*. Other possible operations are left to future research. If $t$ and $u$ are trees, such that the *leftmost non-terminal leaf* of $t$ is equal to the *root* of $u$, then $t \circ u$ is the tree that results from substituting this non-terminal leaf in $t$ by tree $u$. The partial function $\circ$ is called *substitution*. We will write $(t \circ u) \circ v$ as $t \circ u \circ v$, and in general $(..((t_1 \circ t_2) \circ t_3) \circ ..) \circ t_n$ as $t_1 \circ t_2 \circ t_3 \circ ... \circ t_n$.

The restriction *leftmost* in the definition is motivated by the fact that it eliminates different derivations consisting of the same subtrees.

## Parse

Tree $T$ is a *parse* of input string $s$ with respect to a corpus $C$, iff the *yield* of $T$ is equal to $s$ and there are subtrees $t_1,...,t_n \varepsilon C$, such that $T = t_1 \circ ... \circ t_n$. The set of parses of $s$ with respect to $C$, is thus given by:

$$Parses(s,C) = \{T \mid yield(T) = s \wedge \exists t_1,...,t_n \varepsilon C: T = t_1 \circ ... \circ t_n\}$$

The definition correctly includes the trivial case of a subtree from the corpus whose *yield* is equal to the complete input string.

## Derivation

A *derivation* of a parse $T$ with respect to a corpus $C$, is a tuple of subtrees $\langle t_1,...,t_n \rangle$ such that $t_1,...,t_n \varepsilon C$ and $t_1 \circ ... \circ t_n = T$. The set of derivations of $T$ with respect to $C$, is thus given by:

$$Derivations(T,C) = \{\langle t_1,...,t_n \rangle \mid t_1,...,t_n \varepsilon C \wedge t_1 \circ ... \circ t_n = T\}$$

## Probability

**Subtree.** Given a subtree $t_1 \varepsilon C$, a function *root* that yields the root of a tree, and a node labeled $X$, the conditional probability $P(t=t_1 \mid root(t)=X)$ denotes the probability that $t_1$ is substituted on $X$. If $root(t_1) \neq X$, this probability is 0. If $root(t_1) = X$, this probability can be estimated as the ratio between the number of occurrences of $t_1$ in $C$ and the total number of occurrences of subtrees $t'$ in $C$ for which holds that $root(t') = X$. Evidently, $\Sigma_i P(t=t_i \mid root(t)=X) = 1$ holds.

**Derivation.** The probability of a derivation $\langle t_1,...,t_n \rangle$ is equal to the probability that the subtrees $t_1,...,t_n$ are combined. This probability can be computed as the product of the conditional probabilities of the subtrees $t_1,...,t_n$. Let $lnl(x)$ be the leftmost non-terminal leaf of tree $x$, then:

$$P(\langle t_1,...,t_n \rangle) = P(t=t_1 \mid root(t)=S) * \Pi_{i=2 \text{ to } n} P(t=t_i \mid root(t) = lnl(t_{i-1}))$$

**Parse.** The probability of a parse is equal to the probability that any of its derivations occurs. Since the derivations are mutually exclusive, the probability of a parse $T$ is the sum of the probabilities of all its derivations. Let $Derivations(T,C) = \{d_1,...,d_n\}$, then: $P(T) = \Sigma_i P(d_i)$. The conditional probability of a parse $T$ given input string $s$, can be computed as the ratio between the probability of $T$ and the sum of the probabilities of all parses of $s$.

**String.** The probability of a string is equal to the probability that any of its parses occurs. Since the parses are mutually exclusive, the probability of a string $s$ can be computed as the sum of the probabilities of all its parses. Let $Parses(s,C) = \{T_1,...,T_n\}$, then: $P(s) = \Sigma_i P(T_i)$. It can be shown that $\Sigma_i P(s_i) = 1$ holds.

## Monte Carlo Parsing

It is easy to show that an input string can be parsed with conventional parsing techniques, by applying subtrees instead of rules to the input string (Bod, 1992). Every subtree $t$ can be seen as a production rule $root(t) \rightarrow t$, where the non-terminals of the yield of the right hand side constitute the symbols to which new rules/subtrees are applied. Given a polynomial time parsing algorithm, a derivation of the input string, and hence a parse, can be calculated in polynomial time. But if we calculate the probability of a parse by exhaustively calculating all its derivations, the time complexity becomes exponential, since the number of derivations of a parse of an input string grows exponentially with the length of the input string.

Nevertheless, by applying *Monte Carlo techniques* (Hammersley and Handscomb, 1964), we can estimate the probability of a parse and make its error arbitrarily small in polynomial time. The essence of Monte Carlo is very simple: it estimates a probability distribution of events by taking random samples. The larger the samples we take, the higher the reliability. For DOP this means that, instead of exhaustively calculating all parses with all their derivations, we randomly calculate $N$ parses of an input string (by taking random samples from the subtrees that can be substituted on a specific node in the parsing process). The estimated probability of a certain parse given the input string, is then equal to the number of times that parse occurred normalized with respect to $N$. We can estimate a probability as accurately as we want by choosing $N$ as large as we want, since according to the Strong Law of Large Numbers the estimated probability converges to the actual probability. From a classical result of probability theory (Chebyshev's inequality) it follows that the time complexity of achieving a maximum error $\varepsilon$ is given by $O(\varepsilon^{-2})$. Thus the error of probability estimation can be made arbitrarily small in polynomial time - provided that the parsing algorithm is not worse than polynomial.

Obviously, probable parses of an input string are more likely to be generated than improbable ones. Thus, in order to estimate the maximum probability parse, it suffices to sample until stability in the top of the parse distribution occurs. The parse which is generated most often is then the maximum probability parse.

We now show that the probability that a certain parse is generated by Monte Carlo, is exactly the probability of that parse according to the DOP-model. First, the probability that a subtree $t \varepsilon C$ is sampled at a certain point in the parsing process (where a non-terminal $X$ is to be substituted) is equal to $P( t \mid root(t) = X )$. Secondly, the probability that a certain sequence $t_1,...,t_n$ of subtrees that constitutes a derivation of a parse $T$, is sampled, is equal to the product of the conditional probabilities of these subtrees. Finally, the probability that any sequence of subtrees that constitutes a derivation of a certain parse $T$, is sampled, is equal to the sum of the probabilities that these derivations are sampled. This is the probability that a certain parse $T$ is sampled, which is equivalent to the probability of $T$ according to the DOP-model.

We shall call a parser which applies this Monte Carlo technique, a *Monte Carlo parser*. With respect to the theory of computation, a Monte Carlo parser is a probabilistic algorithm which belongs to the class of *B*ounded error *P*robabilistic *P*olynomial time (*BPP*) algorithms. BPP-problems are characterized by the following: it may take exponential time to solve them exactly, but there exists an estimation algorithm with a probability of error that becomes arbitrarily small in polynomial time.

## Experiments on the ATIS corpus

For our experiments we used part-of-speech sequences of spoken-language transcriptions from the Air Travel Information System (ATIS) corpus (Hemphill et al., 1990), with the labeled-bracketings of those sequences in the Penn Treebank (Marcus, 1991). The 750 labeled-bracketings were divided at random into a DOP-corpus of 675 trees and a test set of 75 part-of-speech sequences. The following tree is an example from the DOP-corpus, where for reasons of readability the lexical items are added to the part-of-speech tags.

```
( (S (NP *)
     (VP (VB Show)
         (NP (PP me))
         (NP (NP (PDT all))
             (DT the) (JJ nonstop) (NNS flights)
             (PP (PP (IN from)
                     (NP (NP Dallas)))
                 (PP (TO to)
                     (NP (NP Denver))))
             (ADJP (JJ early)
                   (PP (IN in)
                       (NP (DT the)
                           (NN morning))))))) .)
```

As a measure for *parsing accuracy* we took the percentage of the test sentences for which the maximum probability parse derived by the Monte Carlo parser (for a sample size $N$) is identical to the Treebank parse.

It is one of the most essential features of the DOP approach, that arbitrarily large subtrees are taken into consideration. In order to test the usefulness of this feature, we performed different experiments constraining the *depth* of the subtrees. The depth of a tree is defined as the length of its longest path. The following table

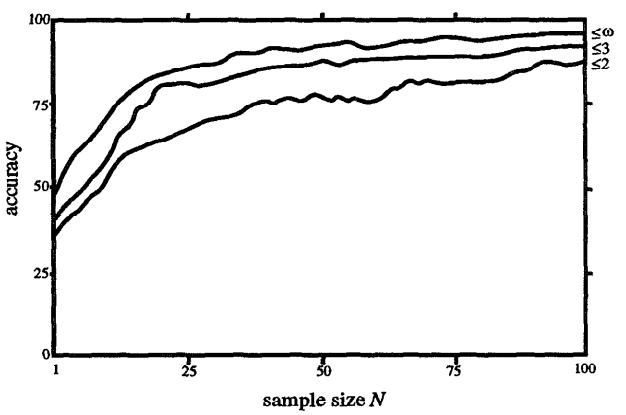
shows the results of seven experiments. The accuracy refers to the parsing accuracy at sample size $N = 100$, and is rounded off to the nearest integer.

| depth | accuracy |
|---|---|
| ≤2 | 87% |
| ≤3 | 92% |
| ≤4 | 93% |
| ≤5 | 93% |
| ≤6 | 95% |
| ≤7 | 95% |
| unbounded | 96% |

Parsing accuracy for the ATIS corpus, sample size $N = 100$.

The table shows that there is a relatively rapid increase in parsing accuracy when enlarging the maximum depth of the subtrees to 3. The accuracy keeps increasing, at a slower rate, when the depth is enlarged further. The highest accuracy is obtained by using all subtrees from the corpus: 72 out of the 75 sentences from the test set are parsed correctly.

In the following figure, parsing accuracy is plotted against the sample size $N$ for three of our experiments: the experiments where the depth of the subtrees is constrained to 2 and 3, and the experiment where the depth is unconstrained. (The maximum depth in the ATIS corpus is 13.)



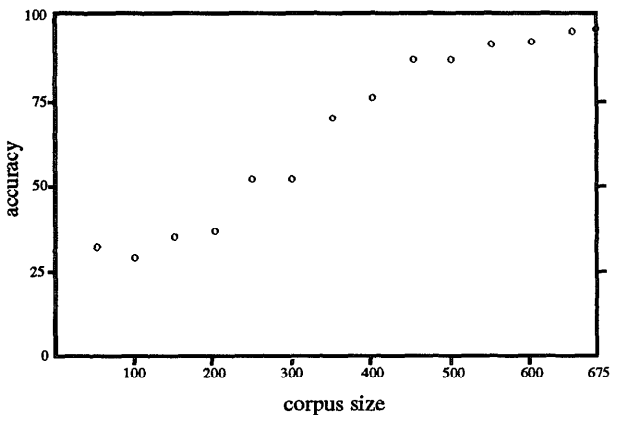

Parsing accuracy for the ATIS corpus, with depth ≤ 2, with depth ≤ 3 and with unbounded depth.

In (Pereira and Schabes, 1992), 90.36% bracketing accuracy was reported using a stochastic CFG trained on bracketings from the ATIS corpus. Though we cannot make a direct comparison, our pilot experiment suggests that our model may have better performance than a stochastic CFG. However, there is still an error rate of 4%. Although there is no reason to expect 100% accuracy in the absence of any semantic or pragmatic analysis, it seems that the accuracy might be further improved. Three limitations of the current experiments are worth mentioning,

First, the Treebank annotations are not rich enough. Although the Treebank uses a relatively rich part-of-speech system (48 terminal symbols), there are only 15 non-terminal symbols. Especially the internal structure of noun phrases is very poor. Semantic annotations are completely absent.

Secondly, it could be that subtrees which occur only once in the corpus, give bad estimations of their actual probabilities. The question as to whether reestimation techniques would further improve the accuracy, must be considered in future research.

Thirdly, it could be that our corpus is not large enough. This brings us to the question as to how much parsing accuracy depends on the size of the corpus. For studying this question, we performed additional experiments with different corpus sizes. Starting with a corpus of only 50 parse trees (randomly chosen from the initial DOP-corpus of 675 trees), we increased its size with intervals of 50. As our test set, we took the same 75 p-o-s sequences as used in the previous experiments. In the next figure the parsing accuracy, for sample size $N = 100$, is plotted against the corpus size, using all corpus subtrees.




Parsing accuracy for the ATIS corpus, with unbounded depth.

The figure shows the increase in parsing accuracy. For a corpus size of 450 trees, the accuracy reaches already 88%. After this, the growth decreases, but the accuracy is still growing at corpus size 675. Thus, we would expect a higher accuracy if the corpus is further enlarged.

## Conclusions and Future Research

We have presented a language model that uses an annotated corpus as a virtual stochastic grammar. We restricted ourselves to substitution as the only combination operation between corpus subtrees. A statistical parsing theory was developed, where one parse can be generated by different derivations, and where the probability of a parse is computed as the sum of the probabilities of all its derivations. It was shown that the maximum probability parse can be estimated as accurately as desired in polynomial time by using Monte Carlo techniques. The method has been succesfully tested on a set of part-of-speech sequences derived from the ATIS corpus. It turned out that parsing accuracy improved if larger subtrees were used.

We would like to extend our experiments to larger corpora, like the Wall Street Journal corpus. This might raise computational problems, since the number of subtrees becomes extremely large. Methods of constraining the number of subtrees, without loosing accuracy, should be investigated. Furthermore, in order to tackle the problem of data sparseness, the possibility of abstracting from corpus data should be included, but statistical models of abstractions of features and categories are not yet available.

**Acknowledgements.** The author is very much indebted to Remko Scha for many valuable comments on earlier versions of this paper. The author is also grateful to Mitch Marcus for supplying the ATIS corpus.

## References

R. Bod, 1992. "A Computational Model of Language Performance: Data Oriented Parsing", *Proceedings COLING'92*, Nantes.

J.M. Hammersley and D.C. Handscomb, 1964. *Monte Carlo Methods*, Chapman and Hall, London.

C.T. Hemphill, J.J. Godfrey and G.R. Doddington, 1990. "The ATIS spoken language systems pilot corpus". *DARPA Speech and Natural Language Workshop*, Hidden Valley, Morgan Kaufmann.

F. Jelinek, J.D. Lafferty and R.L. Mercer, 1990. *Basic Methods of Probabilistic Context Free Grammars*, Technical Report IBM RC 16374 (#72684), Yorktown Heights.

M. Marcus, 1991. "Very Large Annotated Database of American English". *DARPA Speech and Natural Language Workshop*, Pacific Grove, Morgan Kaufmann.

F. Pereira and Y. Schabes, 1992. "Inside-Outside Reestimation from Partially Bracketed Corpora", *Proceedings ACL'92*, Newark.

P. Resnik, 1992. "Probabilistic Tree-Adjoining Grammar as a Framework for Statistical Natural Language Processing", *Proceedings COLING'92*, Nantes.

R. Scha, 1990. "Language Theory and Language Technology; Competence and Performance" (in Dutch), in Q.A.M. de Kort & G.L.J. Leerdam (eds.), *Computertoepassingen in de Neerlandistiek*, Almere: Landelijke Vereniging van Neerlandici (LVVN-jaarboek).

Y. Schabes, 1992. "Stochastic Lexicalized Tree-Adjoining Grammars", *Proceedings COLING'92*, Nantes.