Parse Tree Probability in Data Oriented Parsing*

Remko Bonnema
Paul Buying
Remko Scha

Data oriented parsing systems employ redundant *stochastic tree substitution grammars* (STSGs) to analyse natural language utterances on the basis of an annotated corpus (a *treebank*). An important component of such systems is the way in which the substitution probability of a parse tree fragment is estimated from its occurrences in the treebank. In the standard method for doing this, the probability of a fragment is directly correlated with its occurrence frequency in the collection of all fragments of all corpus trees. We show that this results in undesirable statistical biases. We therefore propose an alternative method, which estimates the substitution probability of a fragment as the probability that it has been involved in the derivation of a corpus tree. We show that this method has more plausible properties.

1 Background

Data oriented parsing (DOP) [Scha, 1990] is a probabilistic approach to natural language interpretation and disambiguation, which differs significantly from previous attempts in this direction. Earlier *probabilistic grammars* were probabilistically enriched competence grammars; the data oriented approach, however, defines a person's language by a stochastic process which recombines structures that are extracted from a representation of the person's past language experience, i.e., a corpus of utterances with syntactic/semantic annotations.

Data oriented language processing is usually implemented as an extremely redundant *stochastic tree substitution grammar* (STSG). The substitutable trees employed by this grammar are simply all the fragments that can be extracted from the corpus. The substitution probability of a tree of a particular category is estimated as the probability of sampling it from the collection of all fragments of the same category that is extracted from the total of all corpus trees [Bod, 1993]. Before we go into detail about this model, we will define some notation and terminology.

*A concise version of this article is published [Bonnema et al., 1999] in P. Dekker and G. Kerdiles (eds.) *Proceedings of the 12th Amsterdam Colloquium*, Institute for Logic, Language and Computation, Amsterdam, The Netherlands, December 1999.

1.1 Preliminaries

A *context free grammar* (CFG) is a four-tuple $G = \langle V, T, R, S\rangle$, where V is the set of variables (non-terminals), T is the set of terminals, R is the set of production rules, and $S \in V$ is the distinguished start symbol. The sets V, T and R are always assumed to be finite. A CFG-derivation of γ from α is a finite sequence of applications of a rule $A \rightarrow \beta$ of R, that transforms α into γ.

We use the *labeled bracketing* notation for a CFG-derivation $\tau_1 \Rightarrow \tau_2 \Rightarrow \cdots \Rightarrow \tau_n$ of τ_n from τ_1 in which each τ_i is written using labeled parentheses. This means that for each derivation step $\tau_i \Rightarrow \tau_j$ using the rule $A \rightarrow \alpha$, τ_j is obtained from rewriting one variable A in τ_i as ${}_A(\alpha)$ (instead of rewriting A as α). Thus, the labeled parentheses are used as auxiliary symbols around the right hand side of applied rules to make the denoted derivation unique. They are not part of the described language itself. For example, a rightmost derivation of $SSSS$ using applications of the rule $S \rightarrow SS$, is written as

$$S \Rightarrow {}_S(SS) \Rightarrow {}_S(S\,{}_S(SS)) \Rightarrow {}_S(S\,{}_S(S\,{}_S(SS))) \tag{1}$$

Using this notation, each τ_i in the derivation corresponds to a *parse tree* (or derivation tree). The string obtained by concatenating all leaf symbols of a parse tree τ is called its *yield*, written as $y(\tau)$. Every parse tree thus denotes a CFG-derivation of its yield.

With respect to a CFG $G = \langle V, T, R, S\rangle$ we call a parse tree τ *lexicalized* if $y(\tau) \in T^*$. If $\alpha_1, \ldots, \alpha_k$ are arbitrary parse trees and $A \in V$, we call the parse tree ${}_A(\alpha_1 \cdots \alpha_k)$ a *fragment tree* or *fragment*. If every α_i is lexicalized, ${}_A(\alpha_1 \cdots \alpha_k)$ is called a *constituent tree* or a *constituent*. For example, the parse trees ${}_S(SS)$, ${}_S(S\,{}_S(SS))$ and ${}_S(S\,{}_S(S\,{}_S(SS)))$ displayed in derivation (1) are all fragments, where S is not, because it is not of the form ${}_A(\alpha_1 \cdots \alpha_k)$. No parse tree displayed in (1) is a constituent because not one of them is lexicalized.

A parse tree τ is said to *start* a parse tree τ' iff there exists a CFG-derivation $\tau = \alpha_1 \Rightarrow \cdots \Rightarrow \alpha_n = \tau'$. If a fragment α starts a parse tree τ we call α an *initial fragment* of τ. For a parse tree τ we define $\sigma(\tau)$ to be the set of all initial fragments of τ. If τ has no initial fragments, we let $\sigma(\tau) = \varnothing$. A parse tree may also *occur* in a parse tree according to the following definition. (1) τ' occurs in τ, if τ' starts τ; (2) τ' occurs in $\tau = {}_A(\alpha_1 \cdots \alpha_i \cdots \alpha_k)$ if τ' occurs in α_i.

As an example, consider the lexicalized parse tree $\tau = {}_S(a\,{}_A(a)\,{}_B({}_A(a)\,{}_B(b)))$. The constituents ${}_A(a)$, ${}_B(b)$, ${}_B({}_A(a)\,{}_B(b))$ and ${}_S(aA\,{}_B(AB))$ all occur in τ. The fragment ${}_B(A\,{}_B(b))$ occurs in τ but is not an initial fragment of τ. The set of initial fragments of τ is given by

$$\begin{aligned}\sigma(\tau) = \{&{}_S(aAB),\ {}_S(a\,{}_A(a)B),\ {}_S(aA\,{}_B(AB)),\ {}_S(a\,{}_A(a)\,{}_B(AB)),\\ &{}_S(aA\,{}_B(A\,{}_B(b))),\ {}_S(a\,{}_A(a)\,{}_B(A\,{}_B(b))),\ {}_S(aA\,{}_B({}_A(a)B)),\\ &{}_S(a\,{}_A(a)\,{}_B({}_A(a)B)),\ {}_S(aA\,{}_B({}_A(a)\,{}_B(b))),\ {}_S(a\,{}_A(a)\,{}_B({}_A(a)\,{}_B(b)))\}\end{aligned}$$

For two parse trees τ' and τ, we define $f(\tau'; \tau)$ to be the number of instances of τ' that occur in τ. If τ' does not occur in τ, we let $f(\tau'; \tau) = 0$. For any parse tree τ we define $h(\tau)$ to be the depth of τ, i.e. the number of edges along the longest path from the root to a leaf symbol of τ. For a fragment or constituent τ we use $r(\tau)$ to denote the (root) label or *category* of τ, such that $r({}_A(\alpha_1 \cdots \alpha_k)) = A$. This implies that fragments and constituents always have an associated category, where parse trees in general might not (i.e. if $\tau \in T$). Let $N(\tau)$ refer to the number of non-root symbols of a constituent τ that are non-terminal.

Throughout this paper we assume that a *corpus* or *treebank* $\mathcal{T}$ is given as a collection of constituent trees $\tau_1, \ldots, \tau_n$. The term *collection* is used to emphasize the property that $\tau_i = \tau_j$ may hold for $i \neq j$. *Corpus* and *treebank* are used interchangeably. Given a treebank $\mathcal{T}$, we define $\mathcal{C}$ to be the set of constituents occurring in $\tau \in \mathcal{T}$. We use $\sigma[\mathcal{C}] = \bigcup_{i=1}^{n} \sigma(\tau_i)$ to refer to the set of all initial fragments of $\mathcal{T}$. We sometimes use a non-terminal $A \in V$ as a subscript of the set $\mathcal{C}$ to denote the restriction of the set to constituents of category A. Analogous to the definition above, $\sigma[\mathcal{C}_A]$ then denotes the set of all initial fragments of $\mathcal{C}_A$.

The total number of instances of a parse tree τ in a corpus $\mathcal{T}$ is defined by $f(\tau; \tau_1, \ldots, \tau_n) = \sum_{i=1}^{n} f(\tau; \tau_i)$. In case it is clear which treebank is meant, $f(\tau; \tau_1, \ldots, \tau_n)$ may be abbreviated to $f(\tau)$. Given a parse tree $\tau \notin T$, the *relative occurrence frequency* or *relative frequency* $F(\tau)$ of τ in $\mathcal{T}$ is defined as $F(\tau) = f(\tau)/f(r(\tau))$.

1.2 The Classical DOP Model

To generate a new sentence from fragments present in the corpus, the DOP model defines the composition operation of *leftmost substitution*, a partial function on pairs of fragments. The composition of fragments α and β, written as $\alpha \circ \beta$ is defined if (and only if) the label of β is identical to the leftmost nonterminal of α. If $\alpha \circ \beta$ is defined, it denotes a copy of α in which a copy of β has been substituted for the leftmost nonterminal of α. For example

$${}_A(B\,{}_C(BD)) \circ {}_B(b) = {}_A({}_B(b)\,{}_C(BD))$$

while the composition ${}_A(B\,{}_C(BD)) \circ {}_D(d)$ is undefined. The requirement to substitute on the *leftmost* nonterminal makes the composition operation unique.

A *leftmost derivation* $\alpha_1 \circ \cdots \circ \alpha_n$ of a constituent τ starts with an initial fragment $\alpha_1 \in \sigma(\tau)$. Then, τ is constructed by repeatedly substituting a fragment α_{k+1} for the leftmost nonterminal of the fragment $(((\alpha_1 \circ \alpha_2) \cdots) \circ \alpha_k)$. For example

$${}_A(BC) \circ {}_B(b) \circ {}_C(D\,{}_E(e)) \circ {}_D(d) = {}_A({}_B(b)\,{}_C({}_D(d)\,{}_E(e)))$$

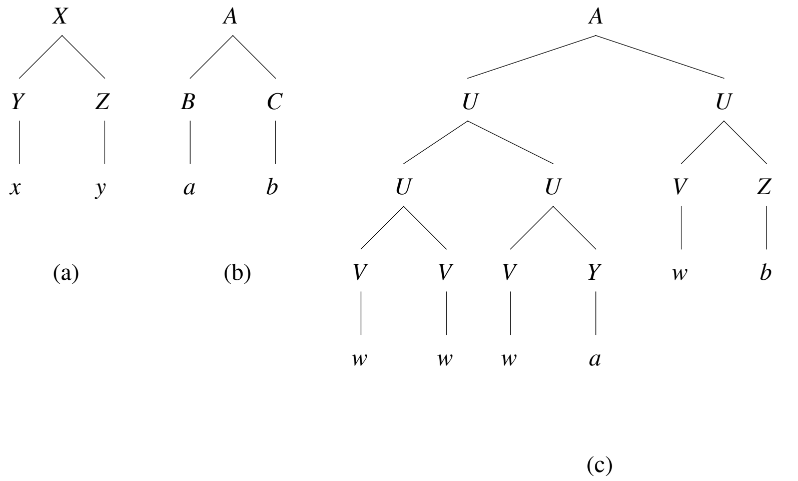
Bod defines the probability of substituting a fragment $\alpha = {}_A(\alpha_1 \cdots \alpha_k)$ for a non-terminal A, as the number of occurrences of α in the treebank, divided by the total number of occurrences of fragments with label A [Bod, 1993; Bod, 1995]. Thus,

$$p(\alpha) = \frac{f(\alpha)}{\sum_{\alpha' \in \sigma[\mathcal{C}_A]} f(\alpha')} \tag{2}$$

For a variable $A \in V$, we will henceforth use $\rho(A) = \sum_{\alpha \in \sigma[\mathcal{C}_A]} f(\alpha)$ to abbreviate the denominator of the right hand side of (2). Given these substitution probabilities, the probability of a derivation $\alpha_1 \circ \cdots \circ \alpha_m = \tau$ of a constituent τ can be computed by taking the product of the probabilities of the substitutions that it consists of:

$$p(\tau) = p(\alpha_1 \circ \cdots \circ \alpha_m) = \prod_{i=1}^{m} p(\alpha_i) \tag{3}$$

The probability of a constituent is equal to the probability that any of its distinct derivations is generated, i.e. the sum of the probabilities of all derivations of that constituent. Let τ be a constituent that is derived from the corpus by derivations $d_1, \ldots, d_n$, where each d_j consists



(a) (b) (c)

Figure 1: Example treebank.

the fragments $\alpha_{1j} \circ \alpha_{2j} \circ \cdots \circ \alpha_{m_j j} = \tau$. Thus, α_{ij} denotes the i-th fragment of derivation j. Then the probability $P(\tau)$ of τ is given by

$$P(\tau) = \sum_{j=1}^{n} \prod_{i=1}^{m_j} p(\alpha_{ij}). \tag{4}$$

1.3 Problems with DOP

The most important innovation that the data oriented approach has brought to stochastic parsing is the decision to use *all* fragments from the corpus directly as a stochastic tree substitution grammar. In this section we question the decision to define the probability of such a fragment as the relative frequency of the fragment among all fragments with the same root label, as in equation (2). For a constituent $\tau = {}_A(\tau_1\tau_2\cdots\tau_k)$, the size of the set of initial fragments is given by the recursive equation $|\sigma(\tau)| = \prod_{i=1}^{k}(|\sigma(\tau_i)| + 1)$. The exponential nature of the fragment extraction operation, implies that large corpus trees make a disproportionately large contribution to the probability mass of the fragments. The biggest constituent of category A that exists in the data, determines the order of magnitude of the probability of all the fragments in that category.

The effect that this property of the model has on the probability which DOP assigns to constituents, can be witnessed even with very small treesizes. Figure 1 shows a toy corpus consisting of three parse trees. The preferred analysis of the string *ab*, given this treebank, should intuitively be 1(b). We will show why this is not the analysis DOP will choose. $\rho(A) = 134$, so *all* fragments of category A get probability $1/134$. $\rho(X) = 4$, on the other hand. This is why ${}_X({}_Y(a)_Z(b))$ is chosen as the preferred analysis of ab. It thus turns out that even the sum of the probabilities of all four ways of deriving the correct answer 1(b), is smaller than the probability assigned to the one possible derivation of ${}_X({}_Y(a)_Z(b))$.

1.4 Constraining the size and form of fragments

Given the problems described above, how was the the DOP model actually used with treebanks of non-trivial size? The answer lies in a simple heuristic: a set of constraints is imposed on the size and form of the fragments that are taken into consideration. Khalil Sima'an [Sima'an, 1999] suggested constraints on four different parameters: maximum depth of a fragment, maximum number of substitution sites in a fragment, maximum number of lexical items, and maximum number of consecutive lexical items in a fragment. The right constraints on the number of substitution sites and (consecutive) terminals compensate for the bias on fragment extraction. To deal with example 1 above, for instance, we only need a constraint on the number of substitution sites to get the desired result. Only a very small percentage of the very large fragments complies with these constraints, while they do not particularly restrict the number of smaller fragments. Reasonable behavior on actual data was achieved with fragments having the following maximum values: two substitution-sites, three consecutive lexical items, nine lexical items in total, and depth four. Experiments showed that lower *and* higher values of the depth parameter caused a decrease in accuracy [Bonnema et al., 1997].

2 A New Probability Model

In this section a new probability model for DOP is proposed, in which the probability of a fragment tree has a closer connection with the extent to which it is supported by occurrances in the treebank. The problems discussed in section 1.3 show that the substitution probability of a fragment (relative to all fragments of the same category), is not proportional to the relative occurrence frequency of the fragment in the treebank. The classical DOP model thus employs a probability measure that invalidates the principle of prefering frequently occurring structures over alternatives that occur less frequent, and should therefore be abolished. Is it possible to maintain the basic ideas behind the data oriented parsing approach, while avoiding the disturbing properties shown above?

2.1 Fragment Probability

Let us reconsider the basic idea behind DOP. We think of every utterance as being generated by a stochastic process which combines fragments by means of the substitution operation. We do not have direct evidence about which fragments people actually use, and with which probabilities. But we can get indirect evidence about this, by (1) collecting a random sample from the population of utterances; and (2) registering a linguist's intuitions about the structure and interpretation of these utterances. Recall that the underlying hypothesis is that the linguist has intuitions about structure that we want to take seriously; but we do not expect her to have reliable introspections about the massively parallel unconscious processes that give rise to these intuitions. The two processes described result in what we call a treebank.

Such a treebank must be transformed into a hypothesis concerning the collection of substitutable fragments and their substitution probabilities. If we stick to DOP's process of constructing new parse trees from fragments occurring in the treebank, the basic structural units for which we want to collect evidence, are the fragments the newly constructed parse tree is composed of. The evidence for these fragments is provided in the form of a collection of annotated utterances in which these fragments might be used. The more a particular fragment is involved as a building block in the corpus trees, the more positive evidence we have for it.

To collect corpus evidence for a single fragment, we should therefore measure the number of times the fragment is *used* in the annotated corpus trees. To do this, we view every corpus tree as the set of all its derivations, each consisting of a sequence of fragment substitutions. The evidence for a fragment supplied by a single constituent is then given by a combination of two factors: the relative frequency of the constituent, and the fraction of the derivations of this constituent that contain a substitution of the fragment. The latter measurement we will call the *fragment distribution* with respect to a particular constituent.

Given a fragment α and a constituent τ, we may define the fragment distribution $\phi(\alpha, \tau)$ as the fraction of all derivations of τ that *start with* α. Let $\delta(\tau)$ denote the set of all possible derivations of a constituent τ. Then,

$$\phi(\alpha, \tau) = \frac{|\{d_j \in \delta(\tau) : \alpha_{1j} \circ \cdots \circ \alpha_{d_j j} = \tau\}|}{|\delta(\tau)|} \tag{5}$$

with $\alpha_{1j} = \alpha$.

For each constituent τ with the same category as the fragment α, we calculate the prior probability that the fragment α is used in a derivation of τ, multiplied by the probability that we select the constituent τ from the treebank. To compute the substitution probability of a fragment on the basis of the whole treebank, we then take the sum of this product over the set of constituents present in the treebank:

$$p'(\alpha) = \sum_{\tau \in \mathcal{C}_A} F(\tau)\phi(\alpha, \tau) \tag{6}$$

In any derivation of τ a non-root variable on τ is either internal to a fragment, or a substitution variable. Since any combination is allowed, the cardinality of $\delta(\tau)$ is equal to the cardinality of the powerset of the set of all non-root variables in τ. Hence, we have $|\delta(\tau)| = 2^{N(\tau)}$. By rewriting equation (5) using the identity $|\delta(\tau)| = 2^{N(\tau)}$, we see that the fragment distribution $\phi(\alpha, \tau)$ is independent of the particular constituent τ. The number of derivations of τ that start with α is given by substracting the available substitution variables of α from the available substitution variables of τ. If τ has $N(\tau)$ substitution variables for which to choose between substituting (for this variable) or not, and an initial fragment α is given, then only $N(\tau) - N(\alpha)$ substitution nodes remain available. Substituting in (5) gives

$$\phi(\alpha, \tau) = \frac{2^{N(\tau)-N(\alpha)}}{2^{N(\tau)}} = \frac{2^{N(\tau)}2^{-N(\alpha)}}{2^{N(\tau)}} = 2^{-N(\alpha)} \tag{7}$$

We define $\phi(\alpha, \tau) = \phi(\alpha) = 2^{-N(\alpha)}$, if $\alpha \in \sigma(\tau)$, and let $\phi(\alpha, \tau) = 0$ for constituents τ such that $\alpha \notin \sigma(\tau)$. Equation (7) shows that the prior probability that a fragment α is used in a derivation of a constituent τ, depends on the complexity of α alone. This is an important property of the proposed probability model.

Since $\phi(\alpha, \tau) = 0$ for constituents τ such that $\alpha \notin \sigma(\tau)$, we restrict the sum over $\mathcal{C}_A$ in equation (6) to constituents τ that are started by α, i.e. for which $\alpha \in \sigma(\tau)$ holds. If $\mathcal{C}_\alpha$ denotes the set of all constituents of $\mathcal{C}$ that are started by α, then $\sum_{\tau \in \mathcal{C}_\alpha} f(\tau) = f(\alpha)$ and therefore $\sum_{\tau \in \mathcal{C}_\alpha} F(\tau) = F(\alpha)$. Thus, taking the sum of the relative frequencies of all constituents that are started by the fragment α, amounts to taking the relative frequency of α itself.

Given a treebank $\mathcal{T}$ and the set $\mathcal{C}$ of constituents occurring in $\mathcal{T}$, we define the probability of a fragment α to be given by the probability function $p' \colon \sigma[\mathcal{C}] \to [0, 1]$, with

$$p'(\alpha) = 2^{-N(\alpha)} F(\alpha) \tag{8}$$

Fragment probability as defined in (8) may be used by an STSG as the probability function on elementary trees. This is permitted only if the sum of p' over all fragments of a particular category is equal to unity. We will consider this in theorem 2.1 on page 8. DOP parsers may employ this distribution, keeping DOPs original combination process of constructing new parse trees from corpus fragments intact. A parse tree probability is then defined as usual. Let τ be a parse tree with n possible derivations $d_1, \ldots, d_n$. Each derivation d_j consists of m_j fragments $\alpha_{1j} \circ \alpha_{2j} \circ \cdots \circ \alpha_{m_j j} = \tau$. The probability $P(\tau)$ of the parse tree τ is given by

$$P(\tau) = \sum_{j=1}^{n} \prod_{i=1}^{m_j} 2^{-N(\alpha_{ij})} F(\alpha_{ij}). \tag{9}$$

2.1.1 Properties of the Fragment Distribution

In this section we will consider the question wether the fragment distribution $\phi(\alpha) = 2^{-N(\alpha)}$ is really a distribution. More specifically, we investigate whether

$$\sum_{\alpha \in \sigma(\tau)} \phi(\alpha) = 1$$

holds for all conceivable constituents τ.

Given a fragment $\alpha \in \sigma(\tau)$ we define the restricted subset $\delta_\alpha(\tau) \subseteq \delta(\tau)$ to be the set of derivations of τ that are started by α, i.e.

$$\delta_\alpha(\tau) = \{d_j \in \delta(\tau) : \alpha_{1j} \circ \cdots \circ \alpha_{d_j j} = \tau \text{ and } \alpha_{1j} = \alpha\}$$

Thus, a fragment α partitions the set of derivations of τ in subsets $\delta_\alpha(\tau)$ of derivations that start with α. We may prove the lemma below, simply by using the distribution of cardinality over the disjoint subsets of this partition, given by

$$|\delta(\tau)| = \left| \bigcup_{\alpha \in \sigma(\tau)} \delta_\alpha(\tau) \right| = \sum_{\alpha \in \sigma(\tau)} |\delta_\alpha(\tau)|$$

Lemma 2.1 *For every constituent tree τ,*

$$\sum_{\alpha \in \sigma(\tau)} 2^{-N(\alpha)} = 1. \tag{10}$$

Proof.

$$\begin{aligned}
\sum_{\alpha \in \sigma(\tau)} 2^{-N(\alpha)} &= \frac{\sum_{\alpha \in \sigma(\tau)} 2^{N(\tau) - N(\alpha)}}{2^{N(\tau)}} = \frac{\sum_{\alpha \in \sigma(\tau)} |\delta_\alpha(\tau)|}{|\delta(\tau)|} \\
&= \frac{|\bigcup_{\alpha \in \sigma(\tau)} \delta_\alpha(\tau)|}{|\delta(\tau)|} = \frac{|\delta(\tau)|}{|\delta(\tau)|} = 1.
\end{aligned}$$

■

2.1.2 Properties of the Fragment Probability Function

The next question of interest is wether the fragment probability $p'(\alpha) = \phi(\alpha)F(\alpha)$ sums to unity over the initial fragments of all constituents of a particular category A, given a fragment distribution function ϕ such that for all τ,

$$\sum_{\alpha \in \sigma(\tau)} \phi(\alpha) = 1 \tag{11}$$

holds. The fragment probability function may be used as the probability function for elementary trees in an STSG, and as a DOP-distribution like in (9), provided this condition is met.

We calculate the sum of the values of p' for all fragments using

$$\sum_{\alpha \in \sigma[\mathcal{C}_A]} \phi(\alpha)F(\alpha) = \sum_{\alpha \in \sigma[\mathcal{C}_A]} \left(\phi(\alpha) \sum_{\tau \in \mathcal{C}_\alpha} F(\tau) \right) \tag{12}$$

which partitions the set $\sigma[\mathcal{C}_A]$ into $2^n - 1$ disjoint subsets. Let $\mathcal{C}_A = \{\tau_1, \ldots, \tau_n\}$, and let T be a set of n-tuples over $\{0, 1\}$, defined as

$$T = \{0, 1\}^n \setminus \langle 0, 0, \ldots, 0 \rangle \tag{13}$$

We use *projection functions* π_i to access the individual elements of tuples. Given arbitrary sets $A_1, A_2, \ldots, A_m$, a projection is defined on the m-ary cartesian product as $\langle a_1, a_2, \ldots, a_m \rangle \mapsto a_i$. Each tuple of T corresponds to a subset of $\sigma[\mathcal{C}_A]$ according to a function $g\colon T \rightarrow 2^{\sigma[\mathcal{C}_A]}$ which is defined as

$$\langle a_1, a_2, \ldots, a_n \rangle \quad \mapsto \quad \bigcap_{a_i = 1} \sigma(\tau_i) \setminus \bigcup_{a_i = 0} \sigma(\tau_i) \tag{14}$$

For a $A \subseteq T$, we let $g[A] = \bigcup_{t \in A} g(t)$. We further define two kinds of groups of tuples. For any k such that $1 \leqslant k \leqslant n$ define

$$T^k = \{t \in T : \sum_{i=1}^{n} \pi_i(t) = k\} \quad \text{and} \quad T(k) = \{t \in T : \pi_k(t) = 1\} \tag{15}$$

The set T^k contains all tuples in which exactly k ones occur. By definition of g this implies that for $t \in T^k$, $g(t)$ consists of fragments shared by exactly k constituents in $\mathcal{C}_A$. Each tuple in T with a one as its k-th element is joined in the set $T(k)$. If $t \in T(k)$ then $g(t) \subseteq \sigma(\tau_k)$. All subsets $g(t)$ with $t \in T(k)$ together form $\sigma(\tau_k)$. Differently put, $g[T(k)] = \sigma(\tau_k)$. The set $\sigma[\mathcal{C}_A]$ is partioned by T in $|T| = 2^n - 1$ disjoint subsets $\bigcup_{t \in T} g(t) = \sigma[\mathcal{C}_A]$ and thus for any two $t, t' \in T$ such that $t \neq t'$ the intersection $g(t) \cap g(t') = \varnothing$.

For the case $n = 3$, for example, with $\mathcal{C}_A = \{\tau_1, \tau_2, \tau_3\}$, the fragment space is partitioned into 7 disjoint subsets, by

$$T = \{\langle 1, 0, 0 \rangle, \langle 0, 1, 0 \rangle, \langle 0, 0, 1 \rangle, \langle 1, 1, 0 \rangle, \langle 1, 0, 1 \rangle, \langle 0, 1, 1 \rangle, \langle 1, 1, 1 \rangle\} \tag{16}$$

The first three tuples in (16) are the elements of T^1, the second three are the elements of T^2, and the last tuple is the only element of T^3.

Theorem 2.1 *Let $\mathcal{C}$ be the set of constituents occurring in a corpus $\mathcal{T}$. Then,*

$$\sum_{\alpha \in \sigma[\mathcal{C}_A]} 2^{-N(\alpha)} F(\alpha) = 1 \tag{17}$$

Proof. We prove the theorem by showing that

$$\sum_{\alpha \in \sigma[\mathcal{C}_A]} 2^{-N(\alpha)} F(\alpha) = \sum_{\tau \in \mathcal{C}_A} F(\tau). \tag{18}$$

using applications of lemma 2.1 (page 7). Let $\mathcal{C}_A = \{\tau_1, \ldots, \tau_n\}$, and let T and g be defined as above. Then

$$\begin{aligned}
&\sum_{\alpha \in \sigma[\mathcal{C}_A]} 2^{-N(\alpha)} F(\alpha) \\
&= \sum_{\alpha \in \sigma[\mathcal{C}_A]} \left(2^{-N(\alpha)} \sum_{\tau \in \mathcal{C}_\alpha} F(\tau) \right) \\
&= \sum_{t \in T^1} \left(\sum_{\alpha \in g(t)} \phi(\alpha) \sum_{i=1}^{n} \pi_i(t) F(\tau_i) \right) + \cdots + \sum_{t \in T^n} \left(\sum_{\alpha \in g(t)} \phi(\alpha) \sum_{i=1}^{n} \pi_i(t) F(\tau_i) \right) \\
&= \sum_{t \in T^1} \sum_{i=1}^{n} \left(\pi_i(t) F(\tau_i) \sum_{\alpha \in g(t)} \phi(\alpha) \right) + \cdots + \sum_{t \in T^n} \sum_{i=1}^{n} \left(\pi_i(t) F(\tau_i) \sum_{\alpha \in g(t)} \phi(\alpha) \right) \\
&= F(\tau_1) \left(\sum_{t \in T(1)} \sum_{\alpha \in g(t)} \phi(\alpha) \right) + \cdots + F(\tau_n) \left(\sum_{t \in T(n)} \sum_{\alpha \in g(t)} \phi(\alpha) \right) \\
&= F(\tau_1) \sum_{\alpha \in g[T(1)]} \phi(\alpha) \quad + \quad \cdots \quad + \quad F(\tau_n) \sum_{\alpha \in g[T(n)]} \phi(\alpha) \\
&= F(\tau_1) \sum_{\alpha \in \sigma(\tau_1)} \phi(\alpha) \quad + \quad \cdots \quad + \quad F(\tau_n) \sum_{\alpha \in \sigma(\tau_n)} \phi(\alpha) \\
&= F(\tau_1) + \cdots + F(\tau_n) \qquad \text{(using lemma 2.1)}
\end{aligned}$$

■

Another property of the probability model is seen by factoring out the fragment distribution. The fragment distribution $\phi(\alpha) = 2^{-N(\alpha)}$ distributes over the composition operation on fragments, so that

$$\phi(\alpha \circ \beta) = \phi(\alpha)\phi(\beta) \tag{19}$$

For an arbitrary derivation d_j of τ such that $\tau = \alpha_{1j} \circ \cdots \circ \alpha_{m_j j}$ it holds that

$$2^{-N(\tau)} = 2^{-N(\alpha_{1j} \circ \cdots \circ \alpha_{m_j j})} = 2^{-\sum_{i=1}^{m_j} N(\alpha_{ij})} = \prod_{i=1}^{m_j} 2^{-N(\alpha_{ij})}. \tag{20}$$

This means that, using (20), expression (9), on page 7, can be rewritten as

$$\begin{aligned}
P(\tau) &= \sum_{j=1}^{n} \prod_{i=1}^{m_j} 2^{-N(\alpha_{ij})} F(\alpha_{ij}) = \sum_{j=1}^{n} \prod_{i=1}^{m_j} 2^{-N(\alpha_{ij})} \prod_{i=1}^{m_j} F(\alpha_{ij}) \\
&= \sum_{j=1}^{n} 2^{-N(\tau)} \prod_{i=1}^{m_j} F(\alpha_{ij}) = 2^{-N(\tau)} \sum_{j=1}^{n} \prod_{i=1}^{m_j} F(\alpha_{ij}) \\
&= \frac{1}{n} \sum_{j=1}^{n} \prod_{i=1}^{m_j} F(\alpha_{ij}) \quad \text{with } n = 2^{N(\tau)}
\end{aligned}$$

This shows that the probability of a parse tree can be viewed as constituting the *average* of products of the relative frequencies of the fragments involved in each possible derivation of the parse tree.

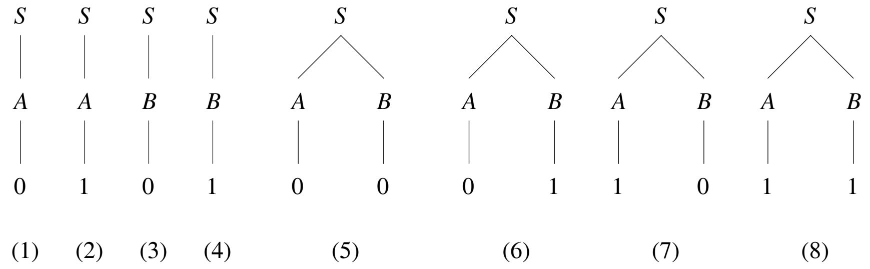


Figure 2: Example treebank, corresponding to PCFG G

2.2 Behavior of the New Probability Model

We now demonstrate how the new DOP model assigns probabilities to trees, and make a comparison with *probabilistic context free grammars* (PCFGs) [Booth and Thompson, 1973] and former DOP models. We demonstrate that the new model is identical to a PCFG model when the independence assumption, made by all PCFG models, is validated by the data. Subsequently, we show with an example how the new model improves on a PCFG when the independence assumption is *not* validated. It will become apparent that the former DOP models did not improve upon PCFG models in this way.

The first example concerns a hypothetical treebank, in which the trees do not exhibit any dependencies between the PCFG rewrite rules that constitute them. The formal definition of such a treebank is given below.

Let G be a PCFG-grammar. With respect to a treebank $\mathcal{T} = \tau_1, \ldots, \tau_n$, the production probabilities of G are given by the standard relative frequency estimator [Chi and Geman, 1998]:$\hat{p}(A \to \alpha) = f(A \to \alpha)/f(A)$. The probability that G assigns to a tree τ, is given by $p(\tau) = \prod_{(A\to\alpha)\in G} p(A \to \alpha)^{f(A\to\alpha;\tau)}$.

Let $\mathcal{T}$ have the property that for all fragments β occurring in $\mathcal{T}$, the following holds.

$$\frac{f(\beta)}{f(r(\beta))} = \prod_{(A\to\alpha)\in G} p(A \to \alpha)^{f(A\to\alpha;\beta)} \tag{21}$$

This equation expresses the proposition that the application of a rewrite rule of G in $\mathcal{T}$, is an independent event. Below, we will call this the *independence constraint*. Figure 2 shows a possible instantiation of $\mathcal{T}$, where G contains the rules $S \to A$ and $S \to B$ with probability 1/4, and $S \to AB$, $A \to 0$, $A \to 1$, $B \to 0$ and $B \to 1$ with probability $1/2$. Figure 3 shows the relative frequencies that are imposed upon the trees by G and the independence constraint.

Given the independence constraint, we can demand of a language model that the probabilities it assigns to trees occurring in the data, equals their observed relative frequency. A probability assignment that deviates from this observed relative frequency, would mean a bias of the model that is not warranted by the data.

By definition of $\mathcal{T}$, a PCFG employing the relative frequency estimator, will assign correct probabilities to all trees in $\mathcal{T}$. Figure 3 shows that the new DOP model assigns correct probabilities to the trees as well. A simple demonstration will show that under the independence

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Rel. freq. of tree	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
PCFG	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
New DOP	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
Former DOP	1/12	1/12	1/12	1/12	1/6	1/6	1/6	1/6

Figure 3: Relative frequencies and probabilities of the trees in figure 2

constraint the new DOP model is always equivalent to a PCFG model, for any choice of G. Let τ have n possible derivations $d_1, \ldots, d_n$. Each derivation d_j consists of m_j fragments $\alpha_{1j} \circ \alpha_{2j} \circ \cdots \circ \alpha_{m_j j} = \tau$. Then,

$$\begin{aligned}
P(\tau) &= 2^{-N(\tau)} \sum_{j=1}^{n} \prod_{i=1}^{m_j} \frac{f(\beta_i)}{f(r(\beta_i))} && \text{by (9)} \\
&= 2^{-N(\tau)} \sum_{j=1}^{n} \prod_{i=1}^{m_j} \prod_{(A\to\alpha)\in G} p(A \to \alpha)^{f(A\to\alpha;\beta_i)} && \text{by (21)} \\
&= 2^{-N(\tau)} \sum_{j=1}^{n} \prod_{(A\to\alpha)\in G} p(A \to \alpha)^{f(A\to\alpha;\tau)} \\
&= \prod_{(A\to\alpha)\in G} p(A \to \alpha)^{f(A\to\alpha;\tau)} && n = 2^{N(\tau)}
\end{aligned}$$

Figure 3 shows that the former DOP model assigns a probability to binary branching trees that is twice as high as the probability assigned to the unary branching ones. In this case, these probabilities are clearly wrong. In the next example we will see how the three models behave if we drop the independence constraint.

Take the trees in figure 2, with their relative frequencies as given in figure 3. We construct a new corpus, by taking all instances of tree (2), and swapping their $_A(1)$ with the $_A(0)$ in all instances of tree (5). This action causes tree (2) to become equal to tree (1), and tree (5) to become equal to tree (7). Our new corpus now consists only of instances of the 6 trees given in figure 4, whose relative frequencies are given in figure 5.

It is obvious that the independence constraint no longer applies to our new treebank. Selection of the rule $_S(A)$ for example should now increase the probability of $_A(0)$ as a continuation of A. Figure 5 gives the probabilities that the different models assign to each of the trees. To the PCFG model the treebank is identical to the previous one, since the relative frequency of rule application did not change. The new DOP model, on the other hand, seems to accurately represent the dependencies in the trees. Take for example the four trees that have an identical relative frequency of $1/8$: trees (2), (3), (5) and (6). Tree (5) has a clear internal dependency. The data show that $_S(AB)$ and $_A(0)$ exhibit a tendency to avoid each other. Tree (6) has the adverse dependency. We see this dependencies reflected in the probabilities assigned to trees (5) and (6), that are respectively $1/64$ below and above the PCFG probabilities.

Constituents (2) and (3) exhibit no internal dependencies between rules. The fact that $_S(B)$ is selected, does not influence the probability of B being rewritten as either 0 or 1. Their assigned probability is therefore equal to the probability assigned by the PCFG model, and thus falls exactly between the values assigned to (5) and (6). The probabilities of the former DOP

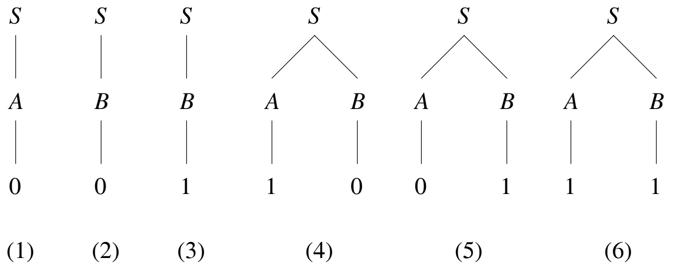


Figure 4: Example of a treebank exhibiting dependencies

	(1)	(2)	(3)	(4)	(5)	(6)
Rel. freq. of tree	1/4	1/8	1/8	1/4	1/8	1/8
PCFG	1/8	1/8	1/8	1/8	1/8	1/8
New DOP	12/64	8/64	8/64	11/64	7/64	9/64
Old DOP	6/48	4/48	4/48	11/48	7/48	9/48

Figure 5: Relative frequencies and probabilities of the trees in figure 4

model also show differences related to internal dependencies, but even greater differences related to difference in size.

3 Computational Issues

The new DOP model can be used to choose the most probable parse tree, given several alternatives. A parsing/disambiguation algorithm using the new model, involves two steps. (1) Creation of a parse forest of an input string, using, for example, a CFG-grammar that is obtained by extracting all CFG-rules from the fragments in the corpus. Many well-known space and time optimalization techniques can be applied. (2) Selection of the most probable analysis, by ranking every tree τ in the parse forest by the value of $P(\tau)$ as assigned by our new DOP.

A real treebank, such as the Penn *Wall Street Journal* Treebank [Marcus et al., 1993], contains 50.000 trees, and covers more than one million words. The number of fragments that can be extracted from these trees, as well as the number of possible derivations for a parse tree of average size, is many orders of magnitude beyond practical computability. To approximate this probability nonetheless, we need to obtain a manageable and representative set of fragments, with a reliable probability assigned to them.

Sampling fragments To approximate the substitution probability of a fragment, we use a sampling algorithm consisting of two steps. The two steps in the sampling process correspond to the two terms in the definition of fragment probability: the relative frequency of the constituent $F(\alpha)$, and the fragment distribution $\phi(\alpha) = 2^{-N(\alpha)}$.

The first step in the sampling algorithm, is to randomly pick a constituent τ from the set of constituents $\mathcal{C}$ in the treebank. The probability of picking τ is determined by its frequency in the treebank. The second step, is to pick an initial fragment of τ. The probability of picking a particular initial fragment α, should correspond to the theoretical prior probability that α starts a derivation of τ, namely $2^{-N(\alpha)}$. This is achieved by first picking a derivation d from the uniformly distributed set of possible derivations $\delta(\tau)$ of τ, and second, taking the first element of d.

These two steps are iterated n times, resulting in a sample of n constituents (sample 1), and n fragments (sample 2). We use $f^1(\tau)$ to denote the frequency of τ in sample 1, and $f^2(\alpha)$ for the frequency of α in sample 2. Note that the frequency of trees of a particular category, $f^1(r(\tau))$, is equal in both samples: Every time a constituent of category A is picked in step 1, an initial fragment of category A is picked in step 2.

We define the estimated probability of a fragment α to be

$$\hat{p}(\alpha) = \frac{f^2(\alpha)}{f^1(r(\alpha))}$$

The term $\hat{p}(\alpha)$ expresses the number of times α was picked relative to the total number of fragments of the same category as α that were picked. This measure must necessarily sum to unity for all fragments of the same category. Below we will show how $\hat{p}(\alpha)$ tends toward $p(\alpha) = F(\alpha)2^{-N(\alpha)}$.

Using $f^1(\tau_\alpha)$ for the sampled frequency of constituents τ starting with α, we can rewrite the definition of $\hat{p}(\alpha)$ as follows:

$$\hat{p}(\alpha) = \frac{f^2(\alpha)}{f^1(r(\alpha))} = \frac{f^1(\tau_\alpha)}{f^1(r(\alpha))}\frac{f^2(\alpha)}{f^1(\tau_\alpha)}$$

The first term in this product is the sampled estimate of $F(\alpha)$, as defined in section 1.1 on page 3. The second term is an estimate of the fragment distribution $\phi(\alpha) = 2^{-N(\alpha)}$. Since each term will converge to the correct relative frequency in the two distributions we sampled from, we have for sample size n:

$$\lim_{n\to\infty} \frac{f^2(\alpha)}{f^1(r(\alpha))} = \lim_{n\to\infty} \frac{f^1(c_\alpha)}{f^1(r(\alpha))}\frac{f^2(\alpha)}{f^1(c_\alpha)} = F(\alpha)2^{-N(\alpha)}$$

Choosing the most probable analysis Note that a manageable set of fragments still does not mean we can distinguish the most probable analysis among all possible parses of a large sentence. Sima'an [Sima'an, 1996] has shown that the problem of computing the most probable parse is not solvable by deterministic polynomial time algorithms. This result applies to both our new and the former DOP models. However, several techniques exist to approximate this value by random sampling. A description of such so-called Monte Carlo techniques can be found in Hammersley et al. [Hammersley and Handscomb, 1964].

4 Conclusion

We have given a detailed demonstration of counterintuitive predictions which the "classical" DOP model generates. The impact that a piece of data has on the predictions of a data oriented parsing system seems to be primarily determined by the sizes of the trees that it occurs in, rather than by its overall occurrence frequency. We proposed an alternative definition of fragment probability that does not suffer from such biases. The measure for the involvement of a fragment in the derivations of a lexicalized tree is shown to be a prior probability depending on the complexity of the fragment alone. We showed how this measure sums to unity over all possible fragments of a lexicalized tree. We used this result to show that the proposed model is a probability function over the set of all fragments of a given category, and consequently, defines an STSG with proper production probabilities. In a detailed example, we demonstrated how the probability of parse trees, as assigned by the new model, compares with those assigned by probabilistic context free grammars and by "classical" DOP models.

References

Bod, R. (1993). Using an annotated corpus as a stochastic grammar. In *Proceedings EACL'93*, Utrecht.

Bod, R. (1995). *Enriching Linguistics with Statistics: Performance Models of Natural Language*. ILLC Dissertation Series 1995-14, University of Amsterdam.

Bonnema, R., Bod, R., and Scha, R. (1997). A DOP model for semantic interpretation. In *Proceedings ACL-EACL 1997*, Madrid, Spain.

Bonnema, R., Buying, P., and Scha, R. (1999). A new probability model for data oriented parsing. In Dekker, P. and Kerdiles, G., editors, *Proceedings of the 12th Amsterdam Colloquium*, Amsterdam, The Netherlands. Institute for Logic, Language and Computation, Department of Philosophy.

Booth, T. and Thompson, R. (1973). Applying probability measures to abstract languages. *IEEE Transactions on Computers*, C-22:442–450.

Chi, Z. and Geman, S. (1998). Estimation of probabilistic context-free grammars. *Computational Linguistics*, 24(2):299–305.

Hammersley, J. and Handscomb, D. (1964). *Monte Carlo Methods*. Chapman and Hall, London.

Marcus, M., Santorini, B., and Marcinkiewicz, M. (1993). Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19:313–330.

Scha, R. (1990). Language Theory and Language Technology; Competence and Performance (in Dutch). In de Kort, Q. and Leerdam, G., editors, *Computertoepassingen in de Neerlandistiek*, Almere: Landelijke Vereniging van Neerlandici (LVVN-jaarboek).

Sima'an, K. (1996). Computational complexity of probabilistic disambiguation by means of tree-grammars. In *Proceedings COLING'96*, Copenhagen, Denmark.

Sima'an, K. (1999). *Learning Efficient Disambiguation*. ILLC Dissertation Series 1999-02, Universiteit Utrecht.

All authors are affiliated with the Department of Computational Linguistics, Institute for Logic, Language and Computation, University of Amsterdam, Spuistraat 134, 1012 VB Amsterdam, The Netherlands.

This research was carried out within the framework of the Priority Programme Language and Speech Technology (TST). The TST-programme is sponsored by NWO (Dutch Organization for Scientific Research).