Journal of Automata, Languages and Combinatorics **u** (v) w, x–y
© Otto-von-Guericke-Universität Magdeburg

A Consistent and Efficient Estimator for Data-Oriented Parsing

ANDREAS ZOLLMANN
School of Computer Science
Carnegie Mellon University, U.S.A.
e-mail: zollmann@cs.cmu.edu

and

KHALIL SIMA'AN
Institute for Logic, Language and Computation
University of Amsterdam, The Netherlands
e-mail: simaan@science.uva.nl

ABSTRACT

Given a sequence of samples from an unknown probability distribution, a statistical estimator aims at providing an approximate guess of the distribution by utilizing statistics from the samples. One crucial property of a 'good' estimator is that its guess approaches the unknown distribution as the sample sequence grows large. This property is called *consistency*.

This paper concerns estimators for natural language parsing under the *Data-Oriented Parsing* (DOP) model. The DOP model specifies how a probabilistic grammar is acquired from statistics over a given training treebank, a corpus of sentence-parse pairs. Recently, Johnson [16] showed that the DOP estimator (called DOP1) is biased and inconsistent. A second relevant problem with DOP1 is that it suffers from an overwhelming computational inefficiency.

This paper presents the first (nontrivial) consistent estimator for the DOP model. The new estimator is based on a combination of held-out estimation and a bias toward parsing with shorter derivations. To justify the need for a biased estimator in the case of DOP, we prove that every non-overfitting DOP estimator is statistically biased. Our choice for the bias toward shorter derivations is justified by empirical experience, mathematical convenience and efficiency considerations. In support of our theoretical results of consistency and computational efficiency, we also report experimental results with the new estimator.

Keywords: statistical parsing, Data-Oriented Parsing, consistent estimator

1. Motivation

A formal grammar describes a set of sentence-analysis pairs, where the analysis is a syntactic construct, often graphically represented as a tree. A major problem with natural language grammars is *ambiguity*, i.e., a linguistic grammar often associates multiple analyses with the same sentence. In contrast, humans usually tend to perceive a single analysis given the context in which the utterance occurs.

Ambiguity resolution in state-of-the-art parsing models is based on probabilistic (also called stochastic) grammars, formal grammars extended with a probabilistic component. The probabilistic component consists of probabilities attached to the grammar productions, and formulae that specify how to calculate probabilities of derivations and analyses in terms of the production probabilities.

A probabilistic grammar thus associates a probability with every sentence-analysis pair. The probabilities allow the ranking of the different analyses associated with the input sentence in order to select the most probable one as the model's best guess of the human preferred analysis. Naturally, for this approach to be effective, the probability values must constitute good approximations of the human disambiguation capacity. Hence, usually the probabilities are estimated from statistics over suitable, representative data.

Most existing parsing models acquire probabilistic grammars from *treebanks*, large bodies of text (corpora) where every sentence is (manually) annotated with the correct syntactic analysis, called *parse tree* (or parse); cf. e.g. [19]. In the *treebank grammars* variant, both the symbolic and the probabilistic components are acquired directly from the treebank, e.g., [2, 10, 11, 25, 6].

Two major decisions are made when acquiring a probabilistic grammar from a treebank: (1) what symbolic grammar to acquire, i.e., what kind of (local) contextual evidence should be encoded in the acquired nonterminals and productions, and (2) how to estimate the probabilities of the grammar productions in order to obtain a probability distribution over sentence-parse pairs that reflects pairs not available in the treebank.

This paper addresses the problem of how to estimate the probabilities for the *fragments* that the Data-Oriented Parsing (DOP) model [22, 2] acquires from a treebank. Informally speaking, a fragment is a subtree (of arbitrary size) of a treebank parse tree. Crucially, the DOP model acquires the multiset of *all* fragments of the treebank parse trees and employs it as a stochastic tree-substitution grammar (STSG) [2].

The original DOP estimator, called the DOP1 estimator [2, 7], was recently proven to be *inconsistent* [16]. Informally, an estimator is consistent if its estimated distribution approaches the actual treebank distribution to any desirable degree when the treebank grows toward infinity. The commonly used maximum-likelihood (ML) estimator, which is known to be consistent, is futile in the case of DOP. Because DOP employs all treebank fragments, the ML estimator will reserve zero probability for parse trees outside the training treebank [8, 27].

We present a new estimator for DOP that combines held-out estimation [18], which is ML estimation over held-out data, with the idea of parsing with the shortest derivation [4]. Intuitively speaking, because the new estimator is based on held-out estimation, it avoids overfitting and retains the consistency of the ML estimator. The shortest derivation property provides an efficient approximation to the complex optimization that the ML estimator involves, without sacrificing consistency. We provide a proof of consistency for the new estimator, and show furthermore that the shortest-derivation approximation constrains the set of treebank fragments considerably, thereby enabling very efficient DOP parsers.

This paper is structured as follows. Section 2 recalls notation, definitions and preliminaries concerning probabilistic grammars and statistical estimation. Section 3 provides an overview of the DOP model and STSGs and describes related work on statistical estimators for DOP. Section 4 shows that every non-overfitting DOP estimator is biased and presents the new estimator in detail. Section 5 provides a proof of consistency of the new estimator, and shows that this estimator results in efficient DOP models. In Section 6, empirical results of crossvalidation experiments are exhibited. Section 7 presents the conclusions and future work.

2. Preliminaries

2.1. Notation

Let V_T and V_N stand for the finite sets of terminals and nonterminals. *Parse trees* (also simply called parses) are defined over these sets as usual, as the formal constructs which are graphically represented by trees of which the non-leaf (also called internal) nodes are labeled with nonterminal symbols and the leaf nodes with terminal symbols. The symbol Ω stands for the set of all parse trees over V_T and V_N. Given a parse tree $t \in \Omega$, $\text{yield}(t) \in V_T^+$ denotes the yield of t (i.e., the sequence of leaves read from left to right)—its corresponding *sentence*. Further, we write $\mathrm{root}(t)$ for the root label of a tree t.

A *treebank* is a finite sequence of $\langle u, t\rangle$ pairs, where $t \in \Omega$ and $u = \text{yield}(t)$. Because the sentence u can be read off from the leaves of the parse tree t, we can simply treat treebanks as sequences of parse trees. As we will see later, a treebank can serve as input data to an *estimator*, which uses these training trees to infer a probability distribution over *all* possible parse trees. Since the order and counts of the input trees are of relevance to the estimator, treebanks are defined as sequences rather than sets or multisets.

The expression $\arg\max_{x \in S} f(x)$ denotes that $x \in S$ for which $f(x)$ is maximal (if a maximum exists on $f(S)$). In cases of ties, x is chosen among the values maximizing f according to some fixed ordering on S.

For sequences or multisets T, we write $|T|$ to denote the length of the sequence or the cardinality of the multiset, respectively. Further, $x \in T$ is defined as true iff x occurs in T and $C(x, T)$ denotes the frequency count (the number of occurrences) of x in T and $\mathit{rf}(x, T) = C(x, T)/|T|$ the relative frequency of x in T.

2.2. Probabilistic syntactic models

A probabilistic parsing model $M = \langle G, p\rangle$ consists of (1) a formal grammar G that generates the set of parses Ω (and utterances), and (2) a probability distribution over the parses Ω represented by its probability mass function $p : \Omega \to [0, 1]$.[1] Let T be a random variable distributed according to p and $U = \text{yield}(T)$ the corresponding random variable over sentences. The aim of parsing is to select for every input sentence $u \in V_T^+$ its most probable parse tree

$$\begin{aligned}\arg\max_{t\in\Omega} \mathbb{P}(T{=}t \mid U{=}u) &= \arg\max_{t\in\Omega} \frac{\mathbb{P}(T{=}t, U{=}u)}{\mathbb{P}(U{=}u)} = \arg\max_{t\in\Omega} \mathbb{P}(T{=}t, U{=}u) \\ &= \arg\max_{t\in\Omega:\text{yield}(t)=u} \mathbb{P}(T{=}t) = \arg\max_{t\in\Omega:\text{yield}(t)=u} p(t)\end{aligned}$$

where the second equality results from the observation that $\mathbb{P}(U = u)$ does not affect the optimization, and the third from the observation that the joint probability of t and u equals the probability of t if $\text{yield}(t) = u$ and zero otherwise.

[1] Since all probability distributions considered in this paper are discrete, we will from now on use the terms *probability distribution* and *probability mass function* interchangeably.

An (ε-free) *context-free grammar* (CFG) is a quadruple $\langle V_N, V_T, S, \mathcal{R}\rangle$, where $S \in V_N$ is the start nonterminal and $\mathcal{R}$ is a finite set of productions of the form $A \rightarrow \alpha$, for $A \in V_N$ and $\alpha \in (V_N \cup V_T)^+$. A CFG is a rewrite system where terms (from $(V_N \cup V_T)^+$) are rewritten into other terms using the (leftmost) substitution operation. Let $A \in V_N$, $w \in V_T^*$ and $\beta \in (V_N \cup V_T)^*$. Rewriting starts from the initial term S (the start symbol) and is iterative. The leftmost nonterminal A in the current term $wA\beta$ is rewritten by substituting the right-hand side of a production $A \rightarrow \alpha \in \mathcal{R}$, thereby leading to the term $w\alpha\beta$. This rewrite step is denoted $wA\beta \overset{A\rightarrow\alpha}{\Longrightarrow} w\alpha\beta$. If the resulting term u consists only of terminal symbols, then u is a sentence of the grammar. The finite sequence of productions that was involved in rewriting the initial term S into sentence u is called a *derivation* of u. The graphical representation of a derivation, where the identities of the productions are left out, is a tree structure, also called a *parse tree* (generated by that derivation). Note that only leftmost derivations are considered and therefore, in CFGs, a parse is generated by exactly one derivation.

Usually, in a generative probabilistic model, a distribution over the set of parse trees is obtained indirectly. A CFG is extended into a *probabilistic CFG* (PCFG) by adding a *weight function* $\pi : \mathcal{R} \rightarrow [0, 1]$ (also referred to as *weight assignment*) over grammar productions. The probabilities of derivations and parse trees are defined in terms of π as follows:

Derivation probability: The probability of a (leftmost) derivation $\langle r_1, \ldots, r_n\rangle$, where $r_i \in \mathcal{R}$, is given by $\prod_{i=1}^{n} \pi(r_i)$.

Parse probability: The probability $p_\pi(t)$ of a parse tree $t \in \Omega$ is defined as the sum of the probabilities of the different derivations that generate t in the grammar.

Since there is a one-to-one mapping between parses and derivations in PCFGs, the definition of parse probability exceeds the PCFG case to the more general case of stochastic tree-substitution grammars (STSGs), which underly the DOP framework.

A desirable requirement on probabilistic generative grammars as PCFGs is that the sum of the probabilities of all parse trees that the grammar generates is smaller than or equal to 1. This is usually enforced by the requirement on π:

$$(1) \quad \forall A \in V_N : \sum_{f \in \mathcal{R}:\ \mathrm{root}(f)=A} \pi(f) = 1\ .$$

Note that if the CFG production $A \rightarrow \alpha$ is viewed graphically as a tree structure, A is its root node label.

2.3. Treebank Grammars and Estimation

The preceding discussion leaves out the issue of how to obtain the grammars and their production probabilities. The current parsing practice is based on the paradigm of treebank grammars [22, 10], which prescribes that both the productions and their probabilities should be acquired from the treebank.

When acquiring a PCFG from a treebank, the treebank trees are viewed as derived by a CFG. Naturally, there is a unique way for decomposing the CFG derivations in the treebank into the CFG productions that they involve. Let $\mathcal{O}_{\mathrm{TB}}$ denote the multiset of the occurrences

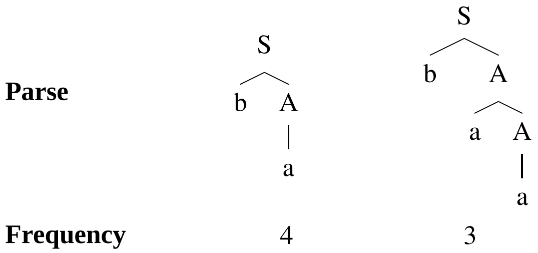

Figure 1: A toy treebank: every parse occurs a number of times equal to its frequency

of the CFG productions in the treebank $\textsc{tb}$. The set of the production rules of the treebank PCFG, denoted $\mathcal{R}_{\textsc{tb}}$, is the set consisting of all unique members of the multiset $\mathcal{O}_{\textsc{tb}}$.

For defining the production weights, let $\mathcal{O}_{\textsc{tb}\restriction A}$ denote the multiset obtained from $\mathcal{O}_{\textsc{tb}}$ by maintaining all and only the production occurrences that have A as root label (i.e., left-hand side). In a treebank PCFG the weight $\pi(A \to \alpha)$ is estimated by $\mathit{rf}(A \to \alpha, \mathcal{O}_{\textsc{tb}\restriction A})$.

Example Let be given the toy treebank $\textsc{tb}$ in Figure 1. Then $\mathcal{R}_{\textsc{tb}} = \{r_1 = S \to b\ A,\ r_2 = A \to a\ A,\ r_3 = A \to a\}$. The frequency counts of the treebank productions are: $C(r_1, \mathcal{O}_{\textsc{tb}}) = 7$, $C(r_2, \mathcal{O}_{\textsc{tb}}) = 3$ and $C(r_3, \mathcal{O}_{\textsc{tb}}) = 7$. Hence, the weight function for this treebank PCFG is given by $\pi(r_1) = 1$, $\pi(r_2) = 0.3$ and $\pi(r_3) = 0.7$.

2.4. *Statistical Estimators and their Properties*

So far, the choice for assigning the production weights for PCFGs according to $\pi(r) = \mathit{rf}\big(r, \mathcal{O}_{\textsc{tb}\restriction \mathrm{root}(r)}\big)$ might seem rather arbitrary to the reader. In general, the preferred assignment is selected using a statistical *estimator* which optimizes some function expressing the fit of the probabilistic grammar to a given treebank $\textsc{tb} = \langle t_1, \ldots, t_n \rangle \in \Omega^n$. Statistical estimation is based on the assumption that there is some *true distribution* over Ω from which all parses in $\textsc{tb}$ were independently sampled. Intuitively speaking, when provided with a treebank, an estimator gives a weight function π which defines a distribution p_π over Ω (the *estimate*) as a guess of the true distribution.

Let $\mathcal{M}_0$ denote the set of all probability distributions over Ω. An *estimator* is a function $\mathit{est} : \Omega^* \to \mathcal{M}_0$ satisfying the condition that for each treebank $\textsc{tb} \in \Omega^*$, the estimate $\mathit{est}(\textsc{tb})$ is in the set $\mathcal{M}_{\textsc{tb}}$ of *eligible* probability distributions over Ω, which we define next.

Let $\Pi_{\textsc{tb}}$ denote the set of all eligible weight functions (π) for the productions of $\textsc{tb}$. (For the specific case of PCFGs, $\Pi_{\textsc{tb}}$ is the set of all functions $\pi : \mathcal{R}_{\textsc{tb}} \to [0,1]$ satisfying Equation (1).) The set of eligible probability distributions $\mathcal{M}_{\textsc{tb}}$ is given by:

$$\mathcal{M}_{\textsc{tb}} = \{p \in \mathcal{M}_0 \ \mid \exists\, \pi \in \Pi_{\textsc{tb}}.\, \forall\, t \in \Omega.\ p(t) = p_\pi(t)\} \ .$$

Remember that $p_\pi(t)$ denotes the parse probability of t resulting from the weight function π.

Maximum-Likelihood and Relative Frequency A common estimator in NLP is the maximum-likelihood (ML) estimator. The ML estimator $\mathit{est}_{\mathrm{ML}} : \Omega^* \to \mathcal{M}_0$ selects the estimate $\mathit{est}_{\mathrm{ML}}(\textsc{tb})$ that maximizes the likelihood of the treebank $\textsc{tb} = \langle t_1, \ldots, t_n \rangle$. Assuming that the parses in TB are independently and identically distributed according to some distribution p allows us to write $p(\textsc{tb}) = \prod_{i=1}^{n} p(t_i)$ for the joint probability of the sequence TB. This leads to the following definition of $\mathit{est}_{\mathrm{ML}}(\textsc{tb})$:

$$\mathit{est}_{\mathrm{ML}}(\textsc{tb}) = \operatorname*{arg\,max}_{p \in \mathcal{M}_{\textsc{tb}}} p(\textsc{tb}) = \operatorname*{arg\,max}_{p \in \mathcal{M}_{\textsc{tb}}} \prod_{i=1}^{n} p(t_i) \ .$$

The ML estimate of a treebank TB need not always exist and need not necessarily be unique. If, however, the relative-frequency distribution $p_{\textsc{tb}}^{rf}$ of the parses in TB, given by $p_{\textsc{tb}}^{rf}(t) = \mathit{rf}(t, \textsc{tb})$, is in $\mathcal{M}_{\textsc{tb}}$, then $\mathit{est}_{\mathrm{ML}}(\textsc{tb})$ exists, is furthermore unique, and equals $p_{\textsc{tb}}^{rf}$ (see, e.g., [21]).

For PCFGs, it turns out that the ML estimate generally will *not* coincide with the relative frequency distribution of the treebank parse trees, but rather with the distribution p_π resulting from the relative frequency estimate π of *productions* from the treebank, which we have encountered in Subsection 2.3. This is the rationale that lies behind the popularity of relative frequency in estimating the weight function π for a treebank PCFG.

Due to the context-freeness assumption of PCFGs, the eligible set of distributions $\mathcal{M}_{\textsc{tb}}$ is often too restricted to contain a distribution that comes close enough to any reasonable distribution underlying natural language treebanks. As we will see in the next section, DOP extends $\mathcal{M}_{\textsc{tb}}$ to encompass any conceivable distribution over the parses in TB.

Bias Based on the expected value of $\mathit{est}(X)$ (denoted $\mathbb{E}\left[\mathit{est}(X)\right]$), *est* is called *biased* for some probability distribution p over Ω if there is an $n \in \mathbb{N}$ such that for the sequence $X = \langle X_1, \ldots, X_n \rangle$ of independent random variables distributed according to p, $\mathbb{E}\left[\mathit{est}(X)\right] \neq p$ holds. Given a set of distributions $\mathcal{M}$, *est* is called *biased w.r.t.* $\mathcal{M}$ if it is biased for some $p \in \mathcal{M}$.

Being unbiased is often considered a quality criterion for an estimator. However, as illustrated e.g. in [14], Section 7.7, for certain problems unbiased estimation is of limited utility.

Consistency Intuitively, an estimator is consistent if it approaches the true distribution assumed to underly the treebank parses when the treebank grows large. In the estimation theory literature, consistency is often defined in terms of an admissible error ε. An estimator is then considered consistent if for each $\varepsilon > 0$, its estimate deviates from the true parameter by more than ε with a probability approaching zero when the sample size approaches infinity. A possible adaption of this view of consistency to our framework of statistical parsing is given by the following definition.

Let $\langle X_i \rangle_{i \in \mathbb{N}}$ be a sequence of independent random variables distributed according to some distribution p over Ω. An estimator *est* is called *consistent for* p if for each real number $\varepsilon > 0$, $\sup_{t \in \Omega} \mathbb{P}\left(|\mathit{est}(\langle X_1, \ldots, X_n \rangle)(t) - p(t)| \geq \varepsilon\right) \to 0$ for $n \to \infty$, i.e.,

$$\lim_{n \to \infty} \sup_{t \in \Omega} \sum_{\substack{\langle t_1, \ldots, t_n \rangle \in \Omega^n : \\ |\mathit{est}(\langle t_1, \ldots, t_n \rangle)(t) - p(t)| \geq \varepsilon}} p(t_1) \cdots p(t_n) = 0 \ .$$

The estimator *est* is called *consistent w.r.t.* $\mathcal{M} \subseteq \mathcal{M}_0$ if it is consistent for each $p \in \mathcal{M}$. An inconsistent estimator will either diverge or it will converge to the wrong distribution. Both situations are not desirable from the point of view of having general models of learning natural language disambiguation.

We note that another common way of defining consistency is in terms of a loss function to approach zero. Such a definition was proposed by Johnson [16]. As shown in [27], an estimator that is consistent in our sense is also consistent in Johnson's sense. Whether the reverse implication holds, is not known to us, but of little relevance in this discussion since we will give a consistency result for our estimator w.r.t. the above definition (implying consistency w.r.t. Johnson's definition), while Johnson proved the inconsistency of DOP1 w.r.t. his definition (implying inconsistency of DOP1 w.r.t. our definition).

ML estimators are typically consistent. This is also the case for the PCFG ML estimator.

3. Overview of Data-Oriented Parsing

Given a training treebank TB, the DOP model acquires from TB a finite set of rewrite productions, called *fragments*, together with their probability estimates. A connected subgraph of a treebank tree t is called a *fragment* iff it consists of one or more context-free productions from t. Figure 4 exemplifies the set of fragments extracted from the treebank of Figure 2.

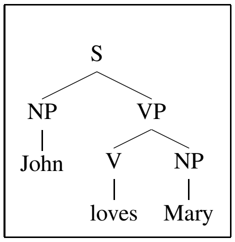

Figure 2: Treebank

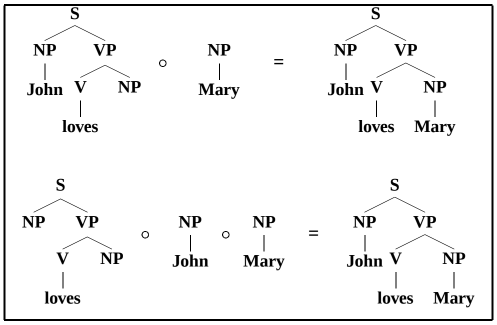

Figure 3: Two different derivations of the same parse

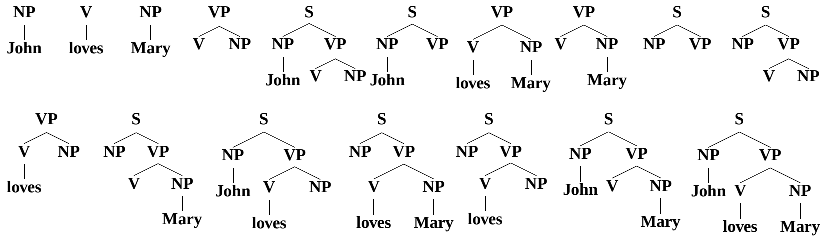

Figure 4: Fragments of the treebank in Figure 2

In DOP, the set of fragments is employed as the set of productions of a tree-substitution grammar (TSG). A TSG is a rewrite system similar to context-free grammars, with the difference that the productions of a TSG are fragments of arbitrary depth. Given a treebank $\textsc{tb}$ of parses over V_N and V_T, the corresponding TSG is a quadruple $\langle V_N,\ V_T,\ S,\ \mathcal{R}_{\textsc{tb}}\rangle$ with start symbol $S \in V_N$ and the finite set $\mathcal{R}_{\textsc{tb}}$ of all fragments of the parse trees in $\textsc{tb}$. Like in CFGs, a (leftmost) derivation in a TSG starts from the start symbol S of the TSG, and proceeds by substituting fragments for nonterminal symbols using the (leftmost) substitution operation (denoted $\circ$). Given fragments $f_1, f_2 \in \mathcal{R}_{\textsc{tb}}$, $f_1 \circ f_2$ is well-defined iff the leftmost nonterminal leaf node μ of f_1 is labeled as the root node of f_2; when well-defined, $f_1 \circ f_2$ denotes the fragment consisting of f_1 with f_2 substituted onto node μ. A sequence $\langle f_1, \ldots, f_n\rangle \in \mathcal{R}^n_{\textsc{tb}}$ such that $\mathrm{root}(f_1) = S$ and $t = (\cdots(f_1 \circ f_2) \circ \cdots) \circ f_n$ is well-defined is called a *derivation* of t. Unlike CFGs, multiple TSG derivations may generate the same parse[2]. For example, the parse in Figure 2 can be derived at least in two different ways as shown in Figure 3.

A Stochastic TSG (STSG) is a TSG extended with a weight function $\pi : \mathcal{R}_{\textsc{tb}} \to [0, 1]$, that underlies the same constraints as in the PCFG case, given by Equation (1). Recall from Subsection 2.2 that for a probabilistic grammar with weight function π, a derivation $d = \langle f_1, \ldots, f_n\rangle$ is assigned the probability $\prod_{i=1}^{n} \pi(f_i)$ and that the parse probability $p_\pi(t)$ is defined as the sum of the probabilities of all derivations that generate parse t.

In analogy to treebank PCFGs, the original DOP model [3], called DOP1, estimates the weight of a fragment $f \in \mathcal{R}_{\textsc{tb}}$ to be equal to its relative frequency among all occurrences of fragments with the same root label in the treebank $\textsc{tb}$, i.e., $\pi(f) = \mathit{rf}\big(f, \mathcal{O}_{\textsc{tb}\restriction\mathrm{root}(f)}\big)$, where we recall that $\mathcal{O}_{\textsc{tb}\restriction A}$ denotes the multiset of all productions from $\textsc{tb}$ with root label A. Johnson [16] shows that the DOP1 distribution p_π, over parse trees, may deviate from the relative frequency distribution $p^{\mathit{rf}}_{\textsc{tb}}$ of the parses in the treebank. In fact, Johnson gives an example showing that even if the treebank constitutes a sample from a parse distribution induced by an STSG, there is no guarantee that the DOP1 estimates approach that distribution as $\textsc{tb}$ grows toward infinity, i.e., the DOP1 estimator is inconsistent.

3.1. DOP and Maximum-Likelihood

Recall from Subsection 2.4 that the set of eligible distributions over parses for an estimator presented with a treebank $\textsc{tb}$ is given by:

$$\mathcal{M}_{\textsc{tb}} = \{p \in \mathcal{M}_0 \ \mid\ \exists\, \pi \in \Pi_{\textsc{tb}}.\, \forall\, t \in \Omega.\ p(t) = p_\pi(t)\}\ .$$

In the case of the DOP model, $\Pi_{\textsc{tb}}$ is the set of all functions $\pi : \mathcal{R}_{\textsc{tb}} \to [0, 1]$ that satisfy Equation (1). Note that $\mathcal{M}_{\textsc{tb}}$ is a superset of the corresponding set of eligible distributions for treebank PCFGs since every CFG production from $\textsc{tb}$ is also a fragment from $\textsc{tb}$ and the conditions on π are identical for PCFGs and STSGs.

Because DOP employs all fragments of a treebank as productions, including the actual parse trees found in the treebank, an interesting situation arises. Any distribution $\tilde{p}$ with $\tilde{p}(t) = 0$ for all parses t that are not in $\textsc{tb}$ is a member of $\mathcal{M}_{\textsc{tb}}$. To see how this happens, assign each fragment f from $\textsc{tb}$ that is also a parse tree in $\textsc{tb}$ the weight $\pi(f) = \tilde{p}(f)$, all other fragments that have the start symbol S as root label the weight zero, and fragments with root labels different from S arbitrary weights subject to Equation (1). It is easy to see that the

[2] Note the difference between parses and fragments: the first are generated, complex events while the latter are atomic rewrite events.

parses in $\textsc{tb}$ are the only parses assigned non-zero parse probabilities by the STSG and that $p_\pi(t) = \tilde{p}(t)$ for all parses $t \in \Omega$.

It follows (by choosing $\tilde{p}(t) = \mathit{rf}(t, \textsc{tb})$) that the relative frequency distribution $p^{rf}_{\textsc{tb}}$ of the $\textsc{tb}$-parses is in $\mathcal{M}_{\textsc{tb}}$. As we know from Subsection 2.4, if $p^{rf}_{\textsc{tb}}$ is in the set of eligible distributions, it is identical to the ML estimate. Unfortunately, $p^{rf}_{\textsc{tb}}$ is not a desirable estimator since only parses occurring in the training treebank are assigned non-zero probabilities and hence the estimator does not generalize. This situation is called *overfitting* in the machine-learning community.

3.2. *Other estimators for DOP*

The fact that DOP1 is inconsistent and that the ML estimator overfits makes consistent estimation for DOP a hard problem. An alternative estimator was introduced in [8], but this estimator is also inconsistent [26].

A newly introduced estimator [26], called Backoff DOP, seems to go most but not all the way towards being consistent. The Backoff DOP estimator is inspired by the known Katz smoothing technique [17] for Markov models over word sequences. Yet, backoff DOP is more complex than Katz backoff for Markov models since it is based on a partial order between fragments (rather than between flat n-grams). The actual implementation described in [26] fails on two points: (1) it starts out from the DOP1 estimates as the initial estimate for the treebank parse trees, which renders that estimator inconsistent, and (2) it employs the estimates as a single model, unlike the way Katz backoff is usually applied, which ruins the statistical properties of Katz backoff.

In the rest of this paper we introduce a new consistent DOP estimator, which allows for more compact DOP models and exhibits improved empirical results over DOP1.

4. A Consistent and Efficient Estimator for DOP

Beside inconsistency, Johnson [16] also showed that the DOP1 estimator is biased. Before developing a new estimator, the question arises whether there exist any unbiased estimators for the DOP model.

4.1. *DOP and Bias*

While the standard DOP maximum-likelihood estimator is unbiased, it is futile because it overfits the treebank. Could any non-overfitting DOP estimator be unbiased? We claim that the answer is no. To prove this claim, we start out with a theorem that provides a sufficient condition for a (general) treebank estimator to be biased.

Theorem 4.1 *Let est* $: \Omega^* \to \mathcal{M}_0$ *be an estimator for which there is a treebank* $\textsc{tb} = \langle t_1, \ldots, t_n \rangle \in \Omega^n$ *and a parse tree* t_0 *outside the treebank (i.e.,* $t_0 \neq t_i$ $(i = 1, \ldots, n)$*) such that est*$(\textsc{tb})(t_0) > 0$. *Then est is biased for each probability distribution* p *over* Ω *that assigns a positive probability to* $\textsc{tb}$ *but probability zero to* t_0*, i.e., for which* $p(t_1) \cdots p(t_n) > 0$ *and* $p(t_0) = 0$.

Proof. Let *est* and $\textsc{tb} = \langle t_1, \ldots, t_n \rangle$ be given as specified above and assume *est* is unbiased for some probability distribution p with $p(t_1) \cdots p(t_n) > 0$ and $p(t_0) = 0$. Let $X_1, \ldots, X_n$ be independent random variables distributed according to p. Then

$$(2) \qquad \mathbb{E}(\mathit{est}(X_1, \ldots, X_n)) = \sum_{\substack{\langle \omega_1, \ldots, \omega_n \rangle \\ \in \Omega^n}} p(\omega_1) \cdots p(\omega_n) \mathit{est}(\omega_1, \ldots, \omega_n) = p.$$

Thus, we have

$$(3) \qquad \sum_{\substack{\omega \in \Omega: \\ p(\omega) \neq 0}} \sum_{\substack{\langle \omega_1, \ldots, \omega_n \rangle \\ \in \Omega^n}} p(\omega_1) \cdots p(\omega_n) [\mathit{est}(\omega_1, \ldots, \omega_n)](\omega) = \sum_{\substack{\omega \in \Omega: \\ p(\omega) \neq 0}} p(\omega).$$

Since $\sum_{\omega \in \Omega:\, p(\omega) \neq 0} p(\omega) = 1$, we obtain from (3):

$$(4) \qquad \sum_{\substack{\omega \in \Omega: \\ p(\omega) \neq 0}} \sum_{\substack{\langle \omega_1, \ldots, \omega_n \rangle \\ \in \Omega^n}} p(\omega_1) \cdots p(\omega_n) [\mathit{est}(\omega_1, \ldots, \omega_n)](\omega) = 1,$$

i.e.,

$$(5) \qquad \sum_{\substack{\langle \omega_1, \ldots, \omega_n \rangle \\ \in \Omega^n}} p(\omega_1) \cdots p(\omega_n) \sum_{\substack{\omega \in \Omega: \\ p(\omega) \neq 0}} [\mathit{est}(\omega_1, \ldots, \omega_n)](\omega) = 1.$$

Since $\sum_{\langle \omega_1, \ldots, \omega_n \rangle \in \Omega^n} p(\omega_1) \cdots p(\omega_n) = 1$ and $\sum_{\omega \in \Omega: p(\omega) \neq 0} [\mathit{est}(\omega_1, \ldots, \omega_n)](\omega) \leq 1$, Equation (5) can only be valid if $\sum_{\omega \in \Omega:\, p(\omega) \neq 0} [\mathit{est}(\omega_1, \ldots, \omega_n)](\omega) = 1$ for all $\omega_1, \ldots, \omega_n \in \Omega$ such that $p(\omega_1) \cdots p(\omega_n) > 0$. But this means $[\mathit{est}(\omega_1, \ldots, \omega_n)](\omega) = 0$ for all $\omega, \omega_1, \ldots, \omega_n \in \Omega$ with $p(\omega) = 0$ and $p(\omega_1) \cdots p(\omega_n) > 0$. Thus, $[\mathit{est}(t_1, \ldots, t_n)](t_0) = 0$, which is a contradiction. □

Now we apply the theorem to DOP. The following corollary states that, given a treebank TB and a DOP estimator that is unbiased w.r.t. the set of eligible distributions $\mathcal{M}_{\textsc{tb}}$, the estimator is bound to overfit the treebank by assigning zero-probabilities to all parse trees outside the corpus.

Corollary 4.2 *Let there be a treebank* $\textsc{tb} \in \Omega^*$ *and a DOP estimator est* $: \Omega^* \rightarrow \mathcal{M}_0$ *that is unbiased w.r.t.* $\mathcal{M}_{\textsc{tb}}$. *Then* $\mathit{est}(\textsc{tb})(t) = 0$ *for all parses* $t \in \Omega$ *that do not occur in* TB.

Proof. Assume indirectly that $\mathit{est}(\textsc{tb})(t_0) > 0$ for some parse tree t_0 that is not in TB. As shown in Subsection 3.1, the relative frequency distribution $p^{\mathit{rf}}_{\textsc{tb}}$ is an instance of $\mathcal{M}_{\textsc{tb}}$. Since $\mathit{rf}(t, \textsc{tb}) > 0$ for all $t \in \textsc{tb}$ and $\mathit{rf}(t_0, \textsc{tb}) = 0$, it follows from Theorem 4.1 that *est* is biased for $p^{\mathit{rf}}_{\textsc{tb}}$. Thus *est* is biased w.r.t. $\mathcal{M}_{\textsc{tb}}$. □

It might be of interest to apply Theorem 4.1 to other estimators in statistical NLP. Note, however, that the theorem is *not* of relevance to probabilistic context free grammars (PCFGs) since for PCFGs, the set of eligible distributions $\mathcal{M}_{\textsc{tb}}$ induced by a treebank TB does not contain a probability distribution that assigns positive probabilities to the trees in TB and zero to an outside tree.

4.2. The New Estimator DOP*

As we have seen in Subsection 3.1, maximum-likelihood estimation in the case of DOP overfits the training treebank. We introduce a new estimator DOP* that is based on the idea of held-out estimation, known from n-gram language modelling. In held-out estimation, the training corpus is randomly split into two parts proportional to a fixed ratio: an *extraction corpus* EC and a *held-out corpus* HC. Applied to DOP, held-out estimation would mean to extract fragments from the trees in EC, but to assign their weights such that the likelihood of the *held-out corpus* HC is maximized. Thus, the set of eligible distributions (cf. Subsection 3.1) is $\mathcal{M}_{\text{EC}}$, from which the estimate that gives the maximum joint probability of the trees in HC is chosen. This way the overfitting problem of standard ML estimation can be avoided.

It can happen that a parse tree in HC is not derivable from the fragments of EC (we will say that it is *not* EC*-derivable*). Therefore, we will actually maximize the joint probability of the EC-derivable trees in HC.

The estimator in the form described so far is problematic: in order to find the best estimate from $\mathcal{M}_{\text{EC}}$ in a reasonable time, expectation-maximization (EM) algorithms such as Inside-Outside [1] would have to be employed. Inside-Outside is a hill-climbing algorithm for statistical parsing. The algorithm starts with an initial weight assignment to grammar productions (in the case of DOP, fragments) and iteratively modifies those weights such that the likelihood of the training corpus increases. Unfortunately, the use of Inside-Outside cannot ensure consistency as it is not guaranteed to (and, in practice, does not [9]) arrive at a global maximum of the likelihood function.

To avoid making use of the EM algorithm, we will make the following simplifying assumption: maximizing the joint probability of the parses in HC is equivalent to maximizing the joint probability of their *shortest derivations*. This assumption turns out handy for several reasons:

- It leads to a closed-form solution for the ML estimate.
- The resulting estimator will only assign non-zero weights to a number of fragments that is linear in the number of depth-1 fragments (i.e., PCFG rules) contained in HC, thereby resulting in an exponential reduction of the number of fragments in the parser. Therefore, the resulting parser is considerably faster than a DOP1 parser.
- The estimator, although not truly maximum likelihood, is consistent.

The assumption also serves a principle of simplicity: a shorter derivation seems a more concise description of a parse tree than a longer one; thus the shortest derivation can be regarded as the preferred way of building up a parse tree from fragments, and the longer derivations as provisional solutions (back-offs) that would have to be used if no shorter ones were available. Furthermore, there are empirical reasons to make the shortest derivation assumption: in [12, 5, 13] it is shown that DOP models that select the preferred parse of a test sentence using the shortest derivation criterion perform very well.

1. Split TB into EC and HC.
2. Extract the fragments from EC.
3. For each parse $t \in$ HC that is derivable from $\mathcal{R}_{\text{EC}}$, pick its shortest derivation.
4. For all fragments $f_1, \ldots, f_N$ involved in the chosen shortest derivations of the parses in HC, determine their frequency counts $r_1, \ldots, r_N$ in these shortest derivations.
5. For the fragments $f_1, \ldots, f_N$, set $\pi(f_j) := \dfrac{r_j}{\sum\limits_{\substack{k\in\{1,\ldots,N\}:\\ \mathrm{root}(f_k)=\mathrm{root}(f_j)}} r_k}$.

   Remove all other fragments from $\mathcal{R}_{\text{EC}}$.

Figure 5: The DOP* estimation algorithm

4.3. Assigning the weights

The algorithm for assigning the fragment weights is stated in Figure 5. We derive this algorithm as the solution for the ML estimate for the EC-derivable trees in HC:

$$(6) \quad \arg\max_{\pi\in\Pi} \prod_{\substack{t\in\text{HC}:\\ t \text{ is EC-derivable}}} [p_\pi(t)]^{C(t,\text{HC})} ,$$

where Π is the set of all $\pi : \mathcal{R}_{\text{EC}} \to [0,1]$ that fulfill the side conditions that for each nonterminal A in EC:

$$(7) \quad \sum_{f\in\mathcal{R}_{\text{EC}}:\, \mathrm{root}(f)=A} \pi(f) = 1 .$$

Under the simplifying assumption indicated above, problem (6) is not affected if each parse probability $p_\pi(t)$ is replaced with the probability of the shortest derivation[3] of t. This leads to the following maximization problem

$$(8) \quad \arg\max_{\pi\in\Pi} \prod_{\substack{t\in\text{HC}:\\ t \text{ is EC-derivable}}} [p^{\text{sh}}_\pi(t)]^{C(t,\text{HC})} ,$$

where $p^{\text{sh}}_\pi(t) = \pi(f_1)\cdots\pi(f_n)$ is the probability of t's shortest derivation $\langle f_1, \ldots, f_n\rangle \in \mathcal{R}^n_{\text{EC}}$. Rearranging the formula and adding together powers of weights of the same fragments ($[\pi(f)]^{e_1}\cdots[\pi(f)]^{e_m} = [\pi(f)]^{e_1+\cdots+e_m}$), we arrive at the term

$$(9) \quad \arg\max_{\pi\in\Pi} [\pi(f_1)]^{r_1}\cdots[\pi(f_N)]^{r_N} ,$$

where $f_1, \ldots, f_N$ are the fragments involved in the shortest derivations of the parses in HC, and r_k is the frequency of fragment f_k in the shortest derivations of all the trees in HC.

Let $\{R_1, \ldots, R_M\}$ be the set of root labels of the fragments $f_1, \ldots, f_N$. Looking back at the side condition (7), we see that each fragment $f \in \mathcal{R}_{\text{EC}} \setminus \{f_1, \ldots, f_N\}$ with

[3] If there are more than one shortest derivation (i.e. many equal-length shortest derivations) for a parse, we can pick any number n of them where in the case that more than one was chosen, each of these derivations is assumed to have derived $1/n$ of the occurrences of that parse, a fraction which needs not necessarily be a whole number.

$\mathrm{root}(f) \in \{R_1, \ldots, R_M\}$ must be assigned the weight $\pi(f) = 0$ in order to maximize the corresponding product in (9). Further, we realize that the weights assigned to fragments $f \in \mathcal{R}_{\text{EC}}$ with $\mathrm{root}(f) \notin \{R_1, \ldots, R_M\}$ have no influence on the outcome of the maximization problem. Since the side conditions for weights of fragments with different roots are independent of each other, we thus obtain an equivalent problem by splitting the product in (9) into a separate optimization problem for every root label $R \in \{R_1, \ldots, R_M\}$ as follows:

$$(10)\qquad \operatorname*{arg\,max}_{\langle \pi(f_j) \rangle_{\mathrm{root}(f_j)=R}} \prod_{\substack{j \in \{1,\ldots,N\}:\\ \mathrm{root}(f_j)=R}} [\pi(f_j)]^{r_j} \ ,$$

where

$$(11)\qquad \sum_{j \in \{1,\ldots,N\}:\, \mathrm{root}(f_j)=R} \pi(f_j) = 1 \ .$$

Thus we have now M optimization problems of the well-known form

$$\operatorname*{arg\,max}_{x_1,\ldots,x_n \in \mathbb{R}} x_1^{c_1} \cdots x_n^{c_n} \ , \text{ where } x_1 + \cdots + x_n = 1$$

with the unique solution [4] for every $i = 1, \ldots, n : x_i = c_i / \sum_{k=1}^{n} c_k$. Applied to our problem, we thus obtain the solutions

$$(12)\quad \pi(f_j) = \frac{r_j}{\displaystyle\sum_{\substack{k \in \{1,\ldots,N\}:\\ \mathrm{root}(f_k)=\mathrm{root}(f_j)}} r_k} \qquad (j = 1, \ldots, N) \ .$$

5. Properties of DOP*

5.1. DOP* *Is Consistent*

We show that DOP* possesses the property of consistency. It turns out that DOP* is not only consistent w.r.t. a set of eligible distributions but even w.r.t. the unrestricted set $\mathcal{M}_0$ of all probability distributions over Ω.

Theorem 5.1 DOP* *is consistent w.r.t. the set* $\mathcal{M}_0$ *of all probability distributions over* Ω.

Proof. Let $\tilde{p}$ be a distribution over Ω and let $\varepsilon > 0$ and $q > 0$ be two real numbers. Further, let $p_{\text{TB}}(t)$ denote the parse probability of t resulting from the DOP* estimator when presented the treebank TB, and recall that $\tilde{p}(\langle t_1, \ldots, t_n \rangle) = \tilde{p}(t_1) \cdots \tilde{p}(t_n)$. In order to show consistency, we will specify an $N \in \mathbb{N}$ such that for each $n \in \mathbb{N}$ with $n \geq N$, we have

$$(13)\quad \forall\, t \in \Omega. \qquad \sum_{\substack{\text{TB} \in \Omega^n:\\ |p_{\text{TB}}(t) - \tilde{p}(t)| \geq \varepsilon}} \tilde{p}(\text{TB}) \leq q$$

and thus

$$\sup_{t \in \Omega} \sum_{\substack{\text{TB} \in \Omega^n:\\ |p_{\text{TB}}(t) - \tilde{p}(t)| \geq \varepsilon}} \tilde{p}(\text{TB}) \leq q \ .$$

[4] A proof of this can for example be found in [20], Subsection 2.4.

To establish (13), we choose a finite set $T \subseteq \Omega$ such that $\sum_{t' \in T} \tilde{p}(t') \geq 1 - \varepsilon/2$ and $\tilde{p}(t') > 0$ for all $t' \in T$. The choice of such a set is possible since $\sum_{t' \in \Omega} \tilde{p}(t') = 1$. In the following, $\mathrm{EC}(\mathrm{TB})$ and $\mathrm{HC}(\mathrm{TB})$ will denote the extraction part and the held-out part of the treebank TB, respectively. Further, let $r \in (0,1)$ be a fixed constant determining the splitting ratio of TB such that $|\mathrm{HC}(\mathrm{TB})| = \lceil r|\mathrm{TB}|\rceil$ and $|\mathrm{EC}(\mathrm{TB})| = |\mathrm{TB}| - \lceil r|\mathrm{TB}|\rceil$.

We will first prove three independent claims:

CLAIM 1 There is an $N_1 \in \mathbb{N}$ such that for all $n \in \mathbb{N}$ with $n \geq N_1$, we have

$$\sum_{\substack{\mathrm{TB}\in\Omega^n:\\ \sum_{t'\in T}\left|\mathit{rf}\left(t',\mathrm{HC}(\mathrm{TB})\right)-\tilde{p}(t')\right|\geq\frac{\varepsilon}{2|T|}}} \tilde{p}(\mathrm{TB}) \qquad \leq q/2\ . \tag{14}$$

Proof. Since T is finite, we can estimate $\sum_{t' \in T} |\mathit{rf}(t', \mathrm{HC}(\mathrm{TB})) - \tilde{p}(t')| \leq |T| \max_{t' \in T} |\mathit{rf}(t', \mathrm{HC}(\mathrm{TB})) - \tilde{p}(t')|$ and thus show that the LHS of (14) fulfills:

$$\sum_{\substack{\mathrm{TB}\in\Omega^n:\\ \exists\, t'\in T.\left|\mathit{rf}\left(t',\mathrm{HC}(\mathrm{TB})\right)-\tilde{p}(t')\right|\geq\frac{\varepsilon}{2|T|^2}}} \tilde{p}(\mathrm{TB}) \quad \leq \quad \sum_{t'\in T} \sum_{\substack{\mathrm{TB}\in\Omega^n:\\ \left|\mathit{rf}\left(t',\mathrm{HC}(\mathrm{TB})\right)-\tilde{p}(t')\right|\geq\frac{\varepsilon}{2|T|^2}}} \tilde{p}(\mathrm{TB})\ .$$

Marginalizing over the EC-portion of TB yields the identical term

$$\sum_{t'\in T} \sum_{\substack{\mathrm{HC}\in\Omega^{\lceil rn\rceil}:\\ \left|\mathit{rf}\left(t',\mathrm{HC}\right)-\tilde{p}(t')\right|\geq\frac{\varepsilon}{2|T|^2}}} \tilde{p}(\mathrm{HC})\ .$$

Applying Chebyshev's inequality to $\mathit{rf}(t', \mathrm{HC})$ with expected value $\tilde{p}(t')$ and variance $\tilde{p}(t')(1 - \tilde{p}(t'))/\lceil rn\rceil \leq 1/(4\lceil rn\rceil)$ yields

$$\sum_{t'\in T} \sum_{\substack{\mathrm{HC}\in\Omega^{\lceil rn\rceil}:\\ \left|\mathit{rf}\left(t',\mathrm{HC}\right)-\tilde{p}(t')\right|\geq\frac{\varepsilon}{2|T|^2}}} \tilde{p}(\mathrm{HC}) \leq \sum_{t'\in T} \frac{1}{4\lceil rn\rceil(\frac{\varepsilon}{2|T|^2})^2} = \frac{|T|^5}{\lceil rn\rceil\varepsilon^2}\ .$$

which yields the desired result when choosing $N_1 = \lceil 2|T|^5/(\varepsilon^2 qr)\rceil$. ◀

CLAIM 2 There is an $N_2 \in \mathbb{N}$ such that for all $n \in \mathbb{N}$ with $n \geq N_2$, we have

$$\sum_{\substack{\mathrm{TB}\in\Omega^n:\\ \exists\, t\in T.\ \ t\text{ occurs in}\\ \mathrm{HC(TB)}\text{ but not in }\mathrm{EC(TB)}}} \tilde{p}(\mathrm{TB}) \leq \frac{q}{2}\ .$$

Proof. Since T is finite and $\tilde{p}(t) > 0$ for all $t \in T$, the probability under $\tilde{p}$ that all $t \in T$ occur in both portions of TB becomes higher than $1 - q/2$ when n gets large enough. This establishes the claim when choosing N_2 large enough. ◀

CLAIM 3 Let TB be a treebank and t a parse tree. Assume that the following inequalities hold:

$$\sum_{t'\in T} |\mathit{rf}(t', \mathrm{HC}(\mathrm{TB})) - \tilde{p}(t')| < \frac{\varepsilon}{2\,|T|}\ \text{, and} \tag{15}$$

$$|p_{\mathrm{TB}}(t) - \tilde{p}(t)| \geq \varepsilon\ . \tag{16}$$

Then there is a $t^* \in T$ such that t^* occurs in $\textsc{hc}(\textsc{tb})$ but not in $\textsc{ec}(\textsc{tb})$.

Proof. Assume indirectly that (15) and (16) hold but that there is no $t^* \in T$ such that t^* occurs in $\textsc{hc}(\textsc{tb})$ but not in $\textsc{ec}(\textsc{tb})$, i.e., that all trees in T that occur in $\textsc{hc}(\textsc{tb})$ also occur in $\textsc{ec}(\textsc{tb})$. Then these trees, in the following denoted by $t_1, \ldots, t_m$, occur also as fragments in $\mathcal{R}_{\textsc{ec}(\textsc{tb})}$. Thus, for each such tree, its shortest derivation from $\mathcal{R}_{\textsc{ec}(\textsc{tb})}$ is the unique length-one derivation consisting only of the tree itself.

Since each derivation of a parse tree in $\textsc{hc}(\textsc{tb})$ contains exactly one fragment whose root label is the start symbol S (namely the first fragment of the derivation), it is easy to see that DOP* assigns each t_j ($j = 1, \ldots, m$) the π-weight (cf. Figure 5, Step 5)

$$(17) \quad \pi(t_j) = \frac{C\left(t_j, \textsc{hc}(\textsc{tb})\right)}{\left|\left\{t' \in \textsc{hc}(\textsc{tb}) \mid t' \text{ is } \textsc{ec}(\textsc{tb})\text{-derivable}\right\}\right|} \geq \mathit{rf}\left(t_j, \textsc{hc}(\textsc{tb})\right) \ .$$

Since parse trees t' not occurring in $\textsc{hc}(\textsc{tb})$ trivially satisfy $\pi(t') \geq \mathit{rf}(t', \textsc{hc}(\textsc{tb}))$, we have for all $t' \in T$

$$p_{\textsc{tb}}(t') \geq \pi(t') \geq \mathit{rf}(t', \textsc{hc}(\textsc{tb})) \ .$$

With (15) (which implies $\forall\, t' \in T.\ |\mathit{rf}(t', \textsc{hc}(\textsc{tb})) - \tilde{p}(t')| < \frac{\varepsilon}{2|T|}$), it follows that

$$(18) \quad \forall\, t' \in T.\, p_{\textsc{tb}}(t') > \tilde{p}(t') - \frac{\varepsilon}{2\,|T|} \ .$$

From this, we can infer for each $t'' \in T$ (by summing up over all $t' \in T \setminus \{t''\}$)

$$\begin{aligned} \sum_{t' \in T \setminus \{t''\}} p_{\textsc{tb}}(t') &> \sum_{t' \in T \setminus \{t''\}} \left(\tilde{p}(t') - \frac{\varepsilon}{2\,|T|}\right) \\ &= \underbrace{\sum_{t' \in T} \tilde{p}(t')}_{\substack{\geq 1 - \varepsilon/2 \\ \text{by Def. of } T}} - \tilde{p}(t'') - \underbrace{(|T| - 1)\frac{\varepsilon}{2\,|T|}}_{\leq \varepsilon/2} \geq 1 - \varepsilon - \tilde{p}(t'') \ . \end{aligned}$$

This means that for all trees $t'' \in T$,

$$\begin{aligned} p_{\textsc{tb}}(t'') = 1 - \sum_{t' \in \Omega \setminus \{t''\}} p_{\textsc{tb}}(t') &\leq 1 - \sum_{t' \in T \setminus \{t''\}} p_{\textsc{tb}}(t') \\ &< 1 - (1 - \varepsilon - \tilde{p}(t'')) = \tilde{p}(t'') + \varepsilon \ . \end{aligned}$$

Together with (18) this yields

$$(19) \quad \forall\, t'' \in T.\ |p_{\textsc{tb}}(t'') - \tilde{p}(t'')| < \varepsilon \ .$$

Now derive from (18), this time by summing up over all $t' \in T$,

$$\sum_{t' \in T} p_{\textsc{tb}}(t') > \sum_{t' \in T} \left(\tilde{p}(t') - \frac{\varepsilon}{2\,|T|}\right) = \sum_{t' \in T} \tilde{p}(t') - |T| \frac{\varepsilon}{2\,|T|} \geq 1 - \varepsilon \ .$$

Thus we have

$$(20) \quad \forall\, t'' \in (\Omega \setminus T).\, p_{\textsc{tb}}(t'') \leq 1 - \sum_{t' \in T} p_{\textsc{tb}}(t') < 1 - (1 - \varepsilon) \leq \tilde{p}(t'') + \varepsilon \ .$$

Further, it holds that

$$\forall\, t'' \in (\Omega \setminus T).\ \tilde{p}(t'') - \varepsilon \le 1 - \underbrace{\sum_{t' \in T} \tilde{p}(t')}_{\substack{\ge 1-\varepsilon/2 \\ \text{by Def. of } T}} - \varepsilon \le -\varepsilon/2 < p_{\text{TB}}(t'')\ ,$$

which together with (20) yields

$$(21)\quad \forall\, t'' \in (\Omega \setminus T).\ |p_{\text{TB}}(t'') - \tilde{p}(t'')| < \varepsilon\ .$$

Inequalities (19) and (21) imply that (16) is false, which is the desired contradiction. ◀

Now we are finally able to specify the required $N \in \mathbb{N}$ such that for all natural numbers $n \ge N$, (13) holds. For that purpose, define $N = \max\{N_1, N_2\}$, where N_1 and N_2 are the numbers provided by Claims 1 and 2, respectively. Then we have for each $t \in \Omega$ and $n \in \mathbb{N}$ with $n > N$,

$$\sum_{\substack{\text{TB} \in \Omega^n: \\ |p_{\text{TB}}(t) - \tilde{p}(t)| \ge \varepsilon}} \tilde{p}(\text{TB}) = \underbrace{\sum_{\substack{\text{TB} \in \Omega^n: \\ \Sigma_{t' \in T} |rf(t', \text{HC(TB)}) - \tilde{p}(t')| \ge \frac{\varepsilon}{2|T|} \\ \text{and } |p_{\text{TB}}(t) - \tilde{p}(t)| \ge \varepsilon}} \tilde{p}(\text{TB})}_{\le q/2 \text{ by Claim 1}} + \underbrace{\sum_{\substack{\text{TB} \in \Omega^n: \\ \Sigma_{t' \in T} |rf(t', \text{HC(TB)}) - \tilde{p}(t')| < \frac{\varepsilon}{2|T|} \\ \text{and } |p_{\text{TB}}(t) - \tilde{p}(t)| \ge \varepsilon}} \tilde{p}(\text{TB})}_{\le \sum\limits_{\substack{\text{TB} \in \Omega^n: \\ \exists\ t^* \in T.\ t^* \text{ occurs in} \\ \text{HC(TB) but not in EC(TB)}}} \tilde{p}(\text{TB}) \text{ by Claim 3}}$$

$$\le \quad q/2 + \underbrace{\sum_{\substack{\text{TB} \in \Omega^n: \\ \exists\ t^* \in T.\ t^* \text{ occurs in} \\ \text{HC(TB) but not in EC(TB)}}} \tilde{p}(\text{TB})}_{\le q/2 \text{ by Claim 2}} \quad \le \quad q\ .$$

□

5.2. *The Number of Extracted Fragments*

The following theorem shows that DOP^* leads to an efficient parser since the number of extracted fragments is linear in the number of nodes in the treebank (as it is the case with PCFG parsers), whereas a DOP1 parser employs an exponential number of such fragments.

Theorem 5.2 *The number of fragments extracted by* DOP^* *is linear in the number of occurrences of depth-one fragments in* HC*, and thus, the number of nodes in* HC.

Proof. For each held-out parse, the estimator extracts fragments from the shortest derivation of that parse. A derivation of a parse tree t has its maximum length when it is built up from the depth-one fragments contained in t. Therefore, the number of fragments extracted from EC for such a derivation is bounded by the number of depth-one fragment occurrences (and hence, the number of nodes) in t. Thus the total number of fragments extracted by DOP^* is bounded by the number of depth-one fragment occurrences (and hence, the number of nodes) in the held-out corpus. □

6. Empirical Results

We exhibit empirical results to support our theoretical findings. The experiments were carried out on the Dutch language OVIS corpus [23], containing 10,049 syntactically and semantically annotated utterances (phrase structure trees). OVIS is a spoken dialogue system for train timetable information. The grammar of the OVIS corpus captures sentences as e.g. "Ik wil niet vandaag maar morgen naar Utrecht" ("I don't want to go today but tomorrow to Utrecht"). Previous experiments on the OVIS corpus have for instance been reported in [24, 26].

Practical Issues To avoid unwanted effects due to specific selection of the held-out corpus, we apply deleted estimation [15]. The present estimator is applied ten times to different equal splits into extraction and held-out portions and the resulting DOP* weight assignments are interpolated together. Note that this does not affect the properties of consistency and linear number of fragments in the number of nodes in the training corpus.

In order to ensure a maximal coverage of the parser, our implementation employs smoothing by discounting some probability mass p_{unkn}, defined as the relative frequency of unknown held-out parses (i.e., parse tree occurrences in HC that are not EC-derivable), and distributing p_{unkn} over the PCFG (depth-one) fragments from TB and the fragments up to depth three of unknown held-out parses. This approach is also consistent [27] as p_{unkn} diminishes when the training corpus becomes large enough.

Testing Unless noted otherwise, the experiments were performed on five fixed random training/test splittings with the ratio 9:1. The figures refer to the average results from these five runs. As common practice on OVIS, one-word utterances were ignored in evaluation as they are easy.

The source codes for training, parsing, and evaluation are publicly available at `http://staff.science.uva.nl/~simaan/dopdis`.

Effects of Inconsistent Estimation We compare DOP* to DOP1 for different maximum-depth constraints on extracted fragments. Figure 6 shows the exact match (EM) rate (number of correctly parsed sentences divided by total number of sentences) for DOP1 and DOP* w.r.t. maximum fragment depth.

Comparing the estimators w.r.t. different levels of fragment depth reveals the influence of consistency on parsing performance: while DOP1 is equivalent to the PCFG estimator for fragment depth one and thus still consistent, this property is increasingly violated as fragments of higher depths are extracted because DOP1 neglects interdependencies of overlapping fragments. The figure is in line with our theoretical explorations earlier in this paper: while DOP*'s performance steadily improves as the fragment depth increases, DOP1 reaches its peak already at depth three and performs even worse when depth-four and depth-five fragments are included. DOP*'s EM rate begins to outperform DOP1's EM rate at depth three.

Efficiency Our tests confirmed the anticipated exponential speed-up in testing time, as Table 1 shows. These data are in line with Figure 7, displaying the number of extracted fragment

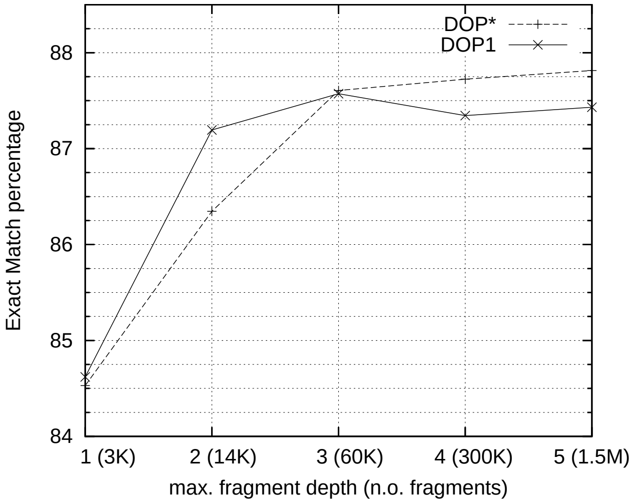

Figure 6: Performance for different maximum-depths of extracted fragments.

types or grammar productions (i.e., counting identical fragments only once) w.r.t. different maximum-depth levels. This number clearly grows exponentially for DOP1, whereas being linearly bounded for DOP*.

Depth	1	2	3	4	5
DOP1	5	6	12	121	1450
DOP*	5	6	6	14	17

Table 1: Parsing time for whole testing corpus in minutes.

7. Conclusions

To the best of our knowledge, the estimator presented in this paper is the first (nontrivial) consistent estimator for the DOP model, and the proof of consistency is the first of its kind concerning probabilistic grammars in computational linguistics. The new estimator solves two major problems for the DOP model simultaneously: (1) the lack of consistent estimators, and (2) the inefficiency caused by the size of the probabilistic grammars that DOP acquires from treebanks. The main solution to both problems comes from a specific preference for shorter derivations as concise descriptions of parse trees.

The current use of the shortest derivation, however, can be easily expanded into a more general consistent estimator that combines the top shortest derivations, i.e., all derivations of length bounded by the length of the shortest-derivation plus some constant. This should improve the coverage of the resulting parser, which could turn out crucial for small treebanks.

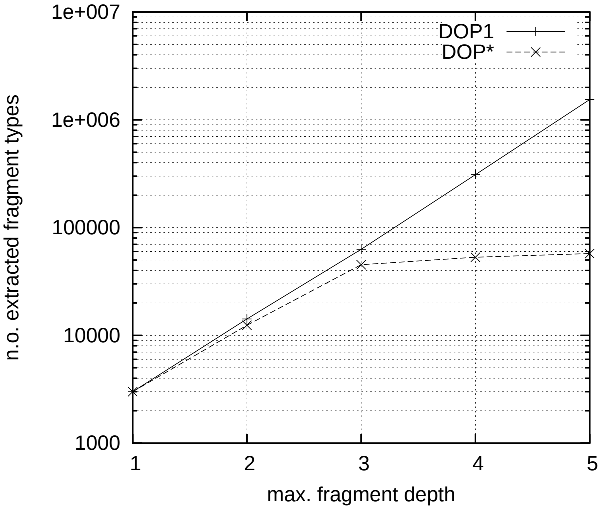

Figure 7: Number of extracted fragment types for different maximum-depth constraints (y-logarithmic scale).

In fact, the choice of a threshold on the shortest derivations could be achieved in a data-driven manner, rather than by fixing it prior to training.

Future empirical work should aim at testing the new estimator on larger corpora in order to establish its empirical merits in comparison with other existing parsers.

References

[1] J. K. Baker. Trainable grammars for speech recognition. In *Proc. of Spring Conference of the Acoustical Society of America*, pages 547–550, 1979.

[2] R. Bod. *Enriching Linguistics with Statistics: Performance models of Natural Language*. PhD dissertation. ILLC dissertation series 1995-14, University of Amsterdam, 1995.

[3] R. Bod. *Beyond Grammar: An Experience-Based Theory of Language*. CSLI Publications, California, 1998.

[4] R. Bod. Parsing with the shortest derivation. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING'2000)*, Saarbrücken, Germany, 2000.

[5] R. Bod. Parsing with the shortest derivation. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING'2000)*, Saarbrücken, Germany, 2000.

[6] R. Bod. What is the minimal set of fragments that achieves maximal parse accuracy?

In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'2001)*, Toulouse, France, 2001.

[7] R. Bod, R. Scha, and K. Sima'an, editors. *Data Oriented Parsing*. CSLI Publications, Stanford University, Stanford, California, USA, 2003. not refereed.

[8] R. Bonnema, P. Buying, and R. Scha. A new probability model for data oriented parsing. In Paul Dekker, editor, *Proceedings of the Twelfth Amsterdam Colloquium*, pages 85–90. ILLC/Department of Philosophy, University of Amsterdam, Amsterdam, 1999.

[9] E. Charniak. *Statistical Language Learning*. MIT Press, Cambridge, MA, 1993.

[10] E. Charniak. Tree-bank Grammars. In *Proceedings AAAI'96*, Portland, Oregon, 1996.

[11] M. Collins. Three generative, lexicalized models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL) and the 8th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 16–23, Madrid, Spain, 1997.

[12] G. De Pauw. Pattern-matching aspects of data-oriented parsing. Presented at Computational Linguistics in the Netherlands (CLIN), Utrecht, 1999.

[13] G. De Pauw. Aspects of Pattern-Matching in DOP. In *Proceedings COLING 2000*, pages 236–242, Saarbrücken, 2000.

[14] M.H. DeGroot. *Probability and statistics*. Addison-Wesley, 2 edition, 1989.

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

[16] M. Johnson. The DOP estimation method is biased and inconsistent. *Computational Linguistics*, 28(1):71–76, 2002.

[17] S.M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing (ASSP)*, 35(3):400–401, 1987.

[18] Ch. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.

[19] M. Marcus, B. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19:313–330, 1993.

[20] H. Ney, S. Martin, and F. Wessel. Statistical language modeling using leaving-one-out. In S. Young and G. Bloothooft, editors, *Corpus-based Methods in Language and Speech Processing*, pages 174–207. Kluwer Academic, Dordrecht, 1997.

[21] D. Prescher, R. Scha, K. Sima'an, and A. Zollmann. On the statistical consistency of DOP estimators. In *Proceedings of the 14th Meeting of Computational Linguistics in the Netherlands*, Antwerp, Belgium, 2004.

[22] R. Scha. Taaltheorie en taaltechnologie; competence en performance. In Q.A.M. de Kort and G.L.J. Leerdam, editors, *Computertoepassingen in de Neerlandistiek, LVVN-jaarboek*, pages 7–22, Almere, The Netherlands, 1990. English translation as: Language Theory and Language Technology; Competence and Performance; http://iaaa.nl/rs/LeerdamE.html.

[23] R. Scha, R. Bonnema, R. Bod, and K. Sima'an. *Disambiguation and Interpretation of Wordgraphs using Data Oriented Parsing*. Technical Report #31, NWO, Priority Programme Language and Speech Technology, `http://grid.let.rug.nl:4321/`, 1996.

[24] K. Sima'an. *Learning Efficient Disambiguation*. PhD dissertation (University of Utrecht). ILLC dissertation series 1999-02, University of Amsterdam, Amsterdam, 1999.

[25] K. Sima'an. Tree-gram Parsing: Lexical Dependencies and Structural Relations. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'00)*, pages 53–60, Hong Kong, China, 2000.

[26] K. Sima'an and L. Buratto. Backoff Parameter Estimation for the DOP Model. In H. Blockeel N. LavraĈ, D. Gamberger and L. Todorovski, editors, *Proceedings of the 14th European Conference on Machine Learning (ECML'03), Lecture Notes in Artificial Intelligence (LNAI 2837)*, pages 373–384, Cavtat-Dubrovnik, Croatia, 2003. Springer. refereed.

[27] A. Zollmann. A Consistent and Efficient Estimator for the Data-Oriented Parsing Model. Master's thesis, Institute for Logic, Language and Computation, University of Amsterdam, Netherlands, May 2004. Available at `http://staff.science.uva.nl/~azollman/publications.html`.