Institute for Logic, Language and Computation

Andreas Zollmann

A Consistent and Efficient Estimator for the Data-Oriented Parsing Model

MoL-2004-02, *received*: May 2004

ILLC Scientific Publications
Series editor: Benedikt Löwe

Master of Logic Thesis (MoL) Series, ISSN: 1387-1951

Institute for Logic, Language and Computation (ILLC)
University of Amsterdam
Plantage Muidergracht 24
NL-1018 TV Amsterdam
The Netherlands
e-mail: illc@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 desired property of an estimator is that its guess approaches the unknown distribution as the sample sequence grows large. Mathematically speaking, this property is called *consistency*.

This thesis presents the first (non-trivial) consistent estimator for the Data-Oriented Parsing (DOP) model. A consistency proof is given that addresses a gap in the current probabilistic grammar literature and can serve as the basis for consistency proofs for other estimators in statistical parsing. The thesis also expounds the computational and empirical superiority of the new estimator over the common DOP estimator *DOP1*: While achieving an exponential reduction in the number of fragments extracted from the treebank (and thus parsing time), the parsing accuracy improves over DOP1.

Another formal property of estimators is *being biased*. This thesis studies that property for the case of DOP and presents the somewhat surprising finding that every unbiased DOP estimator overfits the training data.

Acknowledgements

This thesis could never have been completed without the help of several people.

At the very first I want to thank my advisor Khalil Sima'an, who has been absolutely terrific. He was the one who got me excited about statistical NLP in his course on probabilistic grammars, whose interactive atmosphere had that fascinating *research* smell. During the time of my thesis work, I spent scores of invaluable discussions in his office. Concerning style and legibility, he simply didn't let me get away with anything. Looking back at my first draft of the DOP* chapter, that was more than necessary.

I'd also like to thank Detlef Prescher for his advice on the estimation theory chapter and for reading and commenting on the thesis. Many thanks to Nguyen Thuy Linh as well, who provided me with some very fundamental insights into Data-Oriented Parsing.

I am also grateful to Rens Bod, Dick de Jongh, and Remko Scha for serving on my thesis committee and for their useful comments. Finally, I want to thank all members of the LIT group who put up with watching their mouse pointers move in slow motion over the screen of their paralyzed workstations because the notorious user `azollman` was running `dopdis` again.

Contents

Chapter 1

Introduction

The purpose of *Natural Language Processing (NLP)* is to use computers to automatically analyze human languages. This field of research has applications ranging from speech transcription over text summarization to machine translation. Unlike programming languages, human languages are inherently informal and ambiguous, which makes NLP a challenge that has by far not been mastered to complete satisfaction yet.

This thesis focuses on the task of *sentence parsing*, *i.e.*, calculating the most plausible phrase structure tree (*parse*) for a given sentence (cf. Figure 1.1). Parsing is often the first step of an NLP application: Once the program knows the correct parse tree of a sentence, it can more easily extract characteristic information from it, which may be utilized for semantic analysis or other NLP tasks such as machine translation.

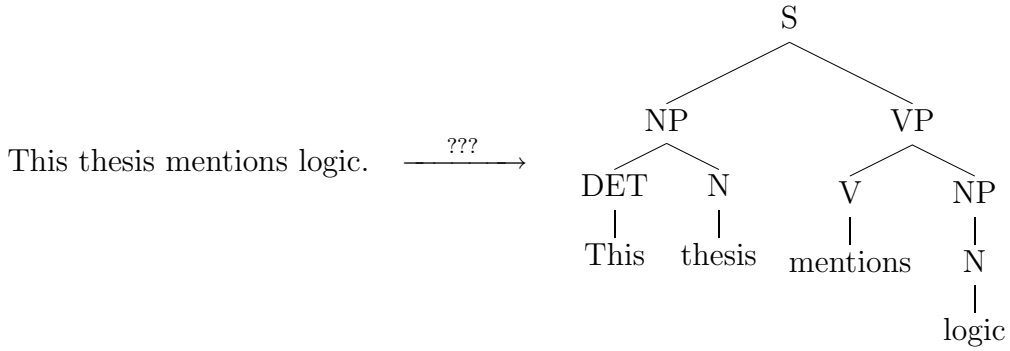


Figure 1.1: Sentence parsing

One reason why natural language processing is so difficult is ambiguity: Besides *semantic ambiguity*—one sentence having different possible meanings—, NLP applications also need to cope with *grammatical ambiguity*: A sentence in

the English language, for instance, has usually many different syntactically correct parses due to different possible ways of attaching prepositional phrases and relative clauses (an example is given in Figure 1.2). Humans resolve ambiguity without great difficulty; often they even fail to notice that different grammatically correct readings of the sentence exist.

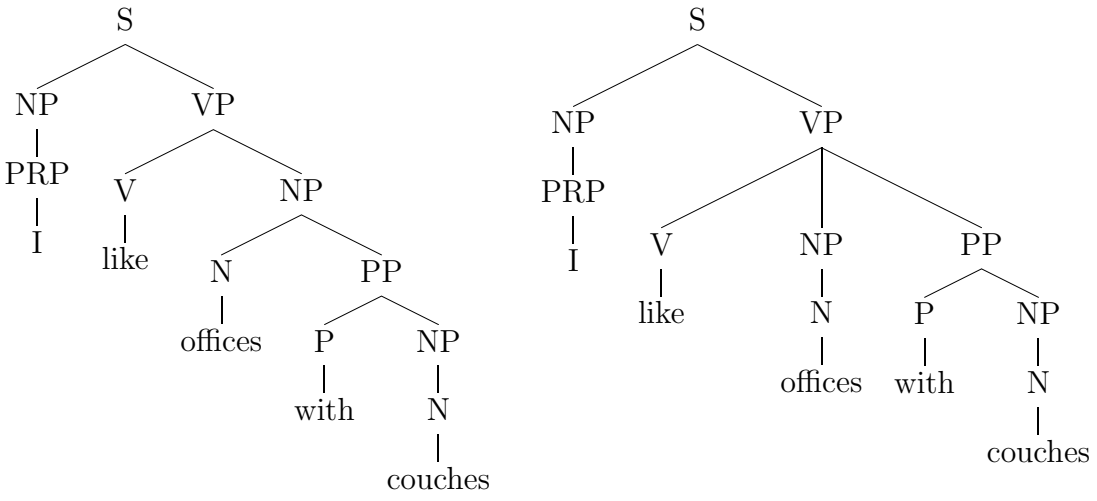


Figure 1.2: Two parses for the sentence "I like offices with couches." The left one is semantically more plausible.

Statistical NLP aims at resolving ambiguities by applying statistical methods to sample data. The idea is to avoid specifying linguistic details (such as the fact that 'table' is a noun) directly in the program code of the NLP application, by having this application learn such details *in context* from a *training corpus*—for instance a treebank consisting of thousands of sample parse trees in the case of a parsing application. By analyzing groups of characters, words, subtrees, or other phenomena in the training data, the NLP application extracts pieces of evidence (*events*) and assigns probabilities to them, which allows ambiguity resolution.

In statistical parsing, the problem of grammatical ambiguity is tackled by assigning degrees of likelihood to parses. The preferred parse for a given sentence is then chosen as the one amongst all grammatically possible parses, that is most likely under the given assignment. Usually, a statistical parser utilizes an underlying probability distribution over the set of possible parse trees, according to which it chooses the most probable parse tree(s) for a given input sentence as the preferred one(s). This probability distribution is often determined by a *stochastic grammar*, consisting of:

1. a formal grammar or rewrite system defining the set of phrase-structure

trees that can be derived from a start symbol 'S' by a successive application of grammar/rewrite rules, and

2. a weight assignment function associating each rule with a real number.

The probability of a parse tree is then given by the weights assigned to the rules of its possible derivations from the stochastic grammar. An example of a stochastic grammar is a *Probabilistic Context-Free Grammar (PCFG)* [Booth, 1969]: Its symbolic backbone is a Context-Free Grammar with rules such as 'S→NP,VP', 'NP→DET N', or 'N→Mary', yielding a language of phrase-structure trees. The weights associated with the rules are real numbers in the interval $[0, 1]$. The probability of a parse tree in the language is now defined as the product of the weights of the rules that were applied to obtain its (left-most) derivation.

In the early stages of statistical parsing, grammars were manually designed by *grammarians* depending on the specific task and domain of the project. The weights of the grammar rules were then obtained automatically from a training corpus, consisting either of natural-language utterances [Baker, 1979, Fujisaki *et al.*, 1989, Lari & Young, 1990, Jelinek *et al.*, 1990] or phrase-structure trees [Pereira & Schabes, 1992]. Corpora of the latter type are also referred to as *treebanks*.

In more recent approaches, even the grammar rules themselves are obtained from the treebank [Scha, 1990, Charniak, 1993]. These so-called *treebank grammars* avoid the costly process of manually designing a grammar and tend to be more robust since they reflect the actual performance of a language user rather than her principal linguistic competence.

This thesis is about *Data-Oriented Parsing (DOP)*[1], a treebank-grammar approach introduced by Remko Scha [Scha, 1990] and formalized by Rens Bod [Bod, 1991]. Its underlying assumption is that human perception of language is based on previous language experiences rather than abstract grammar rules. In the most prominent DOP variants, certain subtrees (called *fragments*) are extracted from the parse trees of the treebank during the training process. These fragments are assigned *weights* between 0 and 1. Fragments can be recombined to parse trees. This way, *new* parse trees—trees that did not necessarily occur in the training corpus—can be obtained. The probability assigned to a parse tree under DOP is determined by the weights of the fragments with which that parse tree can be built up.[2]

[1] Pronounced: 'dopp'.

[2] DOP fits into the framework of stochastic grammars mentioned above. The corresponding class of grammars is called *Stochastic Tree-Substitution Grammars (STSG)* [Bod & Scha, 1996].

Problem statement

Although already achieving state-of-the-art performance, the commonly used model *DOP1* lacks a formal justification such as, *e.g.*, the *maximum-likelihood principle* common in statistical NLP. Furthermore, it has the disturbing property of *inconsistency* [Johnson, 2002]: The probabilities that DOP assigns to parse trees do not converge to the relative frequencies of these trees in the training corpus when that corpus grows large. As we will demonstrate, the failure to fulfill this property causes DOP to incorrectly rank different possible parses for a given sentence based on the evidence in the training corpus. A related problem is the model's bias towards fragments of large parse trees.

Aside from estimation issues, the fact that a parse tree contains an exponential number of DOP fragments in terms of the size (*i.e.*, number of nodes) of the tree has consequences for DOP's computational properties.Whereas PCFG-based models have algorithms for computing the most probable parse in polynomial time w.r.t. sentence length, DOP's case is NP-complete [Sima'an, 1999] and is dominated by the huge size of the grammar.

Contributions

We devise a new DOP model that satisfies the property of consistency. The difficulty lies in the problem that the maximum-likelihood method, according to which the joint probability of the parse trees in the training corpus should be maximized, an estimation method often leading to consistency, is not suitable for DOP: Standard maximum-likelihood estimation results in an *overfitted* (though consistent) estimator that only learns the parses directly occurring in the training data [Bonnema *et al.*, 1999].

In this thesis, we follow a generalized maximum-likelihood approach leading to the first nontrivial consistent estimator for DOP. Our estimator DOP* applies held-out training: Grammar productions are extracted from one part of the training corpus, and their probabilistic weights are assigned based on their participation in derivations of trees from a distinct, *held-out*, part of the corpus. While the formula for those weight assignments is derived from the optimization problem of maximizing the likelihood of the held-out corpus, a simplifying assumption is introduced during the solution process. This assumption leads to a computationally inexpensive closed-form solution and at the same time causes DOP* to prefer learning simpler derivations of parses during the training process.

As a side product of the estimation mechanism, the estimator learns only a number of fragments that is linear in the total number of nodes of the trees in the training corpus, thereby circumventing the inefficiency problems of original DOP without giving up on the idea of using arbitrary-size fragments.[3]

[3]DOP* is not the first efficient estimator for DOP. In [Goodman, 1998], an efficient conver-

Another property of estimators often considered to be desirable is *being unbiased*. We show that DOP estimators cannot—and should not be—completely unbiased. However, we will demonstrate that in contrast to DOP1, DOP* is not biased towards fragments of large parse trees.

Last but not least, we empirically validate our theoretical findings. Using the OVIS corpus to compare the performance of the DOP* parser to DOP1, we find that DOP* achieves better results in parsing accuracy.

1.1 Outline

The following chapter paves the ground for this thesis by acquainting the reader with basic concepts used in statistical parsing. We introduce the Data-Oriented Parsing framework, point out shortcomings of the standard model, and review previous attempts of addressing these shortcomings. Further, the paradigms of *maximum-likelihood estimation* and *held-out estimation* are introduced.

In Chapter 3, we introduce the notions of an estimator, bias, and consistency, and then examine how bias and consistency apply to DOP. We will show that every reasonable DOP estimator must be biased, give a proof that the DOP maximum-likelihood estimator is consistent in preparation for the consistency proof for DOP*, and informally ascertain necessary conditions a consistent DOP estimator must fulfill.

Chapter 4 presents the new estimation procedure DOP*. The chapter also includes a consistency proof for DOP*. Further, we show that DOP* can achieve an exponential reduction in the number of fragments extracted from the training corpus w.r.t. DOP1 and argue that in contrast to DOP1, DOP* is not biased towards fragments of large full parse trees.

Chapter 5 substantiates the theoretical findings of this thesis with empirical evidence. Concluding remarks and directions for possible future work are given in Chapter 6.

sion of DOP1 to a PCFG is presented. The resulting number of PCFG rules is linear in the number of nodes of the training corpus. Although the algorithm is unable to calculate the most probable DOP1 parse, its returned *Labelled Recall Parse* leads to similar parsing accuracy as DOP1. The conversion is only possible for particular DOP estimators, however. Even if an inefficient consistent DOP estimator could be converted into a PCFG estimator in such a way, it is not clear whether Labelled Recall Parsing for that estimator would result in sufficient parsing accuracy since consistency concerns the actual parse tree probabilities.

Chapter 2

Background

As indicated in the introduction, statistical-NLP algorithms extract pieces of evidence (often called: *events*) from a training corpus and assign weights to them based on their frequency in the corpus. A *probabilistic model*[1] then provides a method of combining events to obtain *samples* (*e.g.*, sentences or parse trees) and assigns a probability to each sample based on the weights of the events from which the sample was assembled.

Perhaps the simplest example for a probabilistic model is the *unigram model* over word sequences. Here, the samples are natural language sentences and the events extracted from the training corpus are words and the end-of-sentence mark DOT. The model assigns to each event e its relative frequency $\boldsymbol{rf}\,(e)$ of occurrence in the training corpus. The probability of a sentence (regarded as a sequence $\langle w_1, \ldots, w_n, \text{DOT}\rangle$, where $n \in \mathbb{N}$ and $w_1, \ldots, w_n$ are word-events[2]) is now simply defined as the product of the weights of its events:

$$P_{\text{UNIG}}(\langle w_1, \ldots, w_n, \text{DOT}\rangle) = \boldsymbol{rf}\,(w_1) \cdots \boldsymbol{rf}\,(w_n)\,\boldsymbol{rf}\,(\text{DOT})\ .$$

It is easy to show that P_{UNIG} is a probability distribution over the set of possible sentences.

A probabilistic model for *statistical parsing* uses parse trees (cf. Figure 1.1) as samples. In the case of Data-Oriented Parsing, events are subtrees that can be combined using a *substitution operation*.

This chapter starts out with some basic concepts used in statistical parsing and establishes the notation used throughout this thesis (Section 2.1). Subsequently, we introduce the Data-Oriented Parsing framework, advert to shortcomings of the standard model, and review previous attempts of addressing these shortcomings (Section 2.2). In the remaining sections, two methods of assigning

[1] Not to be confused with *probability models* as introduced in Section 2.3.

[2] Note that the tuple $\langle \text{DOT}\rangle$ also counts as a sentence.

weights to events commonly used in statistical NLP, *maximum-likelihood estimation* and *held-out estimation*, are introduced to prepare the ground for the subsequent chapters.

2.1 Terminology

In this section, we introduce notation that will be employed throughout this thesis.

Sentences and trees

In statistical NLP, the basic units considered are usually the *words* of a fixed natural language. A *sentence* is formally treated as a sequence of words.

When writing about *trees* in this thesis, we mean *phrase structure trees*, that is, trees whose non-leaf nodes are phrase-structure categories (*e.g.*, 'S', 'NP', 'N') and whose leaves are either categories or words. Sometimes, the words of a phrase structure tree are also referred to as *terminals* and the categories as *nonterminals*, alluding to the terminology of Formal Language Theory. A tree with root node 'S' (the *start nonterminal*) all of whose leaves are words and in which 'S' does not occur in any non-root node is called a *full parse tree* (also: *parse tree*, or simply *parse*). A *parse tree for a given sentence* s is a full parse tree whose yield (*i.e.*, its sequence of leaves traversed from left to right) is the sentence s. Confer Figures 1.1 and 1.2 for examples of full parse trees.

Sequences

When talking about sequences, we mean *finite* sequences. The symbol $\circ$ denotes the composition operation for sequences, $|s|$ the length of a sequence s. When using set operators such as $\in, \cap, \cup$, etc., on a sequence, they refer to the induced set of all the elements occurring in the sequence. For a sequence s, we write $\boldsymbol{Count}_s(e)$ for the number of occurrences of the element e in s, and $\boldsymbol{rf}_s(e) = \frac{\boldsymbol{Count}_s(e)}{|s|}$ for e's relative frequency in s. We extend those definitions to sets E of elements by defining $\boldsymbol{Count}_s(E) := \sum_{e \in E} \boldsymbol{Count}_s(e)$ and $\boldsymbol{rf}_s(E) := \sum_{e \in E} \boldsymbol{rf}_s(e)$.

Finally, given a set S, the *star* of S is defined as

$$S^* := \bigcup_{i \in \mathbb{N}} S^i \ .$$

Probability distributions over parse trees

Wherever we come across probability distributions in this thesis, their underlying sample space will be the set **Parses** of all full parse trees. Since this set is countable, we can characterize a probability distribution over **Parses** by its probability function, a function from **Parses** to $[0, 1]$ assigning to each elementary event $\{t\}$ (where $t \in$ **Parses**) its probability, and will from now on sloppily talk about probability distributions $P :$ **Parses** $\rightarrow [0, 1]$.

Corpora and experimental methodology

A *treebank* is a sequence of full parse trees, which are assumed to be independent samples from **Parses** according to a certain probability distribution. Note that this view—fundamental to DOP1 as well as DOP*—completely neglects inter-sentence dependencies such as discourse phenomena.

For experiments, the treebank is split into a *training* and a *testing corpus*. The former is used during the training process, while the latter is a reserved portion of the treebank—not seen during training—, whose sentences (extracted from the leaves of the parse trees) are fed into the parser in order to compare its proposed parse with the parse tree the test sentence was attached to. This practice leads to objective performance figures, which make different parsers comparable to each other.

Estimation

The training procedure in statistical parsing results in a probability distribution over **Parses**, according to which the parser can determine the preferred parse for an input sentence. We will call this procedure an *estimator*. Intuitively, an estimator tries to approximate the 'true' probability distribution assumed to underly the training corpus. In the case of DOP, the estimator is the procedure that assigns weights to fragments and thereby probabilities to full parse trees dependent on the training corpus. An estimator is *consistent* if its estimated probability distribution converges to the 'true' distribution when the training corpus grows large. We will give a formal definition of an estimator and its properties in Chapter 3.

2.2 Data-Oriented Parsing

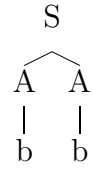
In this section, we introduce the general framework used by current DOP estimators and introduce the DOP1 model. The DOP framework originally set out is actually even more general. A detailed discussion can be found in

[Scha, 1990, Bod & Scha, 1996, Bod, 1998]. In Subsections 2.2.3 and 2.2.4, we advert to shortcomings of the standard model. Subsection 2.2.5 gives an overview and assessment of previous attempts of addressing these shortcomings.

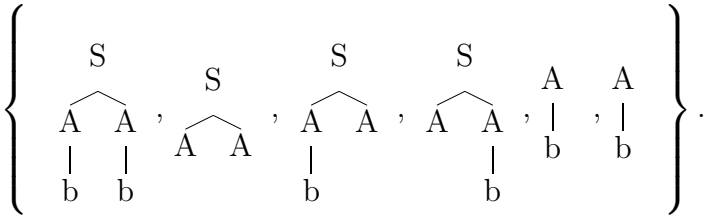
2.2.1 The General Framework

As already mentioned in the introduction, during the training process of DOP, *fragments* (also simply called *subtrees*) are extracted from the full parse trees of the training corpus. For each full parse tree t, the multiset of fragments of t is the multiset of all occurrences of subgraphs f of t, such that

1. f consists of more than one node,
2. f is connected,
3. each non-leaf node in f has the same daughter nodes as the corresponding node in t.

For instance, the fragment multiset of the tree 

is



For a given training corpus $\textsc{tc} = \langle t_1, \ldots, t_N \rangle$, its *fragment corpus* $\mathbf{Frag}_{\textsc{tc}}$ is the composition of the sequences $s_1, \ldots, s_N$, where s_i is the sequence resulting of arranging the elements of the fragment multiset of t_i according to some fixed order. We denote the set induced by $\mathbf{Frag}_{\textsc{tc}}$ as $\mathbf{FragSet}_{\textsc{tc}}$. Using the convention introduced in Section 2.1, we will avoid referring to that set explicitly wherever possible. A fragment $f \in \mathbf{Frag}_{\textsc{tc}}$ is called a *proper fragment w.r.t.* TC if it does not occur as a full parse tree in TC.

In the next step of the DOP estimation procedure, each fragment $f \in \mathbf{Frag}_{\textsc{tc}}$ is assigned a *weight* $\pi(f) \in [0, 1]$ such that the weight assignment to fragments with a common root R forms a probability distribution for each R occurring as a root in $\mathbf{Frag}_{\textsc{tc}}$. In other words, for each such R, π must fulfill the condition

$$\sum_{f:\, \mathrm{root}(f)=R} \pi(f) = 1.$$

As mentioned earlier, DOP allows for the recombination of fragments to new full parse trees. This is done by *tree composition*. The composition of tree t_1 and tree t_2, resulting in the tree $t_1 \circ t_2$, is defined iff the leftmost leaf of t_1 that is a nonterminal is identical to the root of t_2. If defined, $t_1 \circ t_2$ yields a copy of t_1 whose leftmost nonterminal leaf has been replaced by the whole tree t_2. Note that this composition operation is technically not associative.[3] When writing $t_1 \circ t_2 \circ t_3$, we mean $(t_1 \circ t_2) \circ t_3$.

A sequence $\langle f_1, \ldots, f_n \rangle \in (\mathbf{Frag}_{\text{TC}})^n$ such that $t = f_1 \circ \cdots \circ f_n$ is a full parse tree is called a *derivation of* t. The *DOP probability of a derivation* $d = \langle f_1, \ldots, f_n \rangle$ is the product of the weights of the involved fragments:

$$P_{\text{DOP}}(d) := \prod_{i=1}^{n} \pi(f_i)$$

The property that weights of fragments with the same root sum up to one ensures that P_{DOP} is a probability distribution over the set of all derivations.[4] The *DOP probability of a full parse tree* is now simply defined as the sum of the DOP probabilities of all its derivations. The *DOP probability of a sentence* in turn is the sum of the DOP probabilities of its full parse trees. Finally, the *preferred parse(s)* according to DOP is/are the full parse tree(s) with maximal DOP probability.

2.2.2 The DOP1 Estimator

The DOP1 estimator is obtained by choosing the weight assignment π to fragments such that each fragment $f \in \mathbf{Frag}_{\text{TC}}$ is assigned its relative frequency amongst all occurrences of fragments with the same root in $\mathbf{Frag}_{\text{TC}}$:

$$\pi(f) := \frac{\boldsymbol{Count}_{\mathbf{Frag}_{\text{TC}}}(f)}{\boldsymbol{Count}_{\mathbf{Frag}_{\text{TC}}}(\{f' \in \mathbf{Frag}_{\text{TC}} \mid \text{root}(f') = \text{root}(f)\})}$$

A toy example of a training corpus, its resulting fragment corpus, the weights assigned by DOP1, and the calculation of the DOP1 probabilities of some resulting full parse trees is given in the following subsection.

2.2.3 DOP1 Is Inconsistent

DOP1 suffers from some severe problems. One of them is the *inconsistency* of the estimation method. Consider a training corpus in which only the two parse

[3] However, we have $(t_1 \circ t_2) \circ t_3 = t_1 \circ (t_2 \circ t_3)$ if both sides of the equation are defined.

[4] As it is the case for Probabilistic Context Free Grammars (PCFGs), certain recursive DOP grammars 'leak' probability mass such that the derivation probabilities sum up to less than one. We will ignore this issue here.

trees drawn in Figure 2.1 occur, both with equal frequencies. The trees t_1 and t_2 have the same structure and differ merely in the categories assigned to the words "a" and "b". Intuitively, neither of them should be preferred as a parse for the sentence "a b."

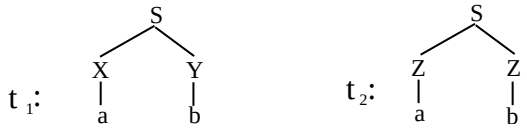


Figure 2.1: Two analyses for the sentence "a b"

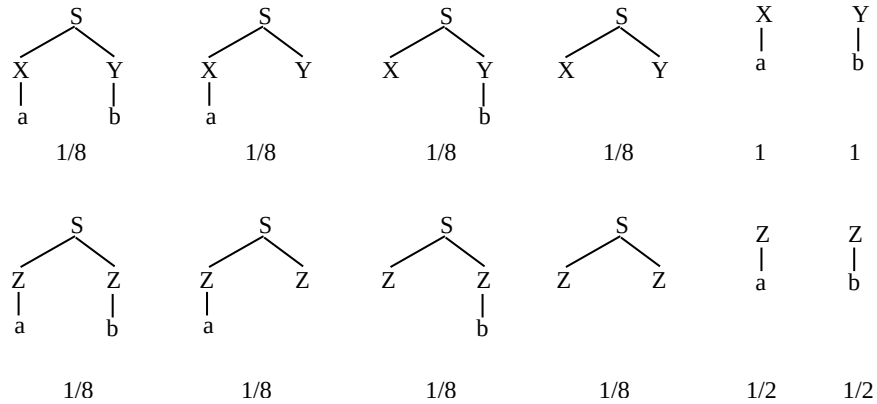


Figure 2.2: The fragments extracted from the "a b" training corpus and their DOP1-weights

Figure 2.2 shows the fragments of the resulting fragment corpus and their weights assigned by DOP1. In order to calculate the DOP1 probabilities of t_1 and t_2 (as can easily be seen, they are the only parses for "a b"), we need to determine all of their derivations and their respective DOP1 probabilities. This has been done in Figure 2.3. Here and in the following, we use the convention to denote a derivation of a full parse tree by the tree itself, in which the nodes at which the compositions of fragments occurred are marked.

Now we can determine the DOP1 probabilities of t_1 and t_2 by summing up the probabilities of their derivations:

$$\begin{aligned} P_{\text{DOP1}}(t_1) &= \tfrac{1}{8} + \tfrac{1}{8} + \tfrac{1}{8} + \tfrac{1}{8} = .5 \\ P_{\text{DOP1}}(t_2) &= \tfrac{1}{8} + \tfrac{1}{16} + \tfrac{1}{16} + \tfrac{1}{32} = .28125 \end{aligned} \tag{2.1}$$

DOP1 thus has an unjustifiably strong preference for one of the two parses of "a b".

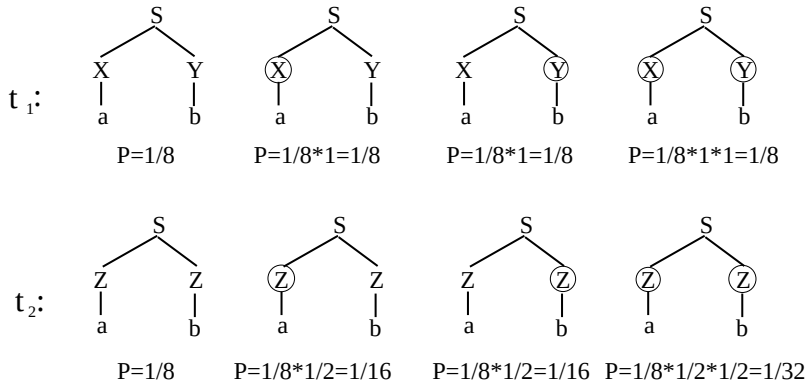


Figure 2.3: The derivations of t_1 and t_2 and their probabilities

Let us now assume that the trees in the training corpus were sampled according to an underlying probability distribution P with $P(t_1) = P(t_2) = 0.5$.[5] Even when the size of the training corpus sampled according to P grows very large (and thus, the relative frequencies of t_1 and t_2 will be roughly 0.5), DOP1 will keep assigning them probabilities close to the ones in Equation 2.1. Intuitively, the DOP estimator should choose the weights it assigns to the fragments in such a way that the resulting probabilities assigned to the derivable full parse trees approach the relative frequencies of their occurrences in the training data when its size grows large. This requirement is made precise by the notion of consistency, which will be formally introduced in Chapter 3. The inconsistency of DOP1 was first shown in [Johnson, 2002].

2.2.4 DOP1 Is Biased Towards Fragments of Large Parse Trees

The problem of DOP1's inconsistency is related to another one: Its bias towards fragments of large full parse trees, illustrated in [Bonnema *et al.*, 1999]. For a tree

$$t = \begin{array}{c} \mathrm{R} \\ \triangle \\ t_1 \cdots t_k \end{array},$$

where $t_1, \ldots, t_k$ are the subtrees under the root R, the size $s_{\mathrm{IF}}(t)$ of the multiset of its initial fragments (that is, the fragments with the same root R), is recursively given by

$$s_{\mathrm{IF}}(t) = \prod_{i=1}^{k} \left(s_{\mathrm{IF}}(t_i) + 1 \right).$$

[5] Note that this distribution results from a DOP estimator assigning weights 0.5 to both *fragments* t_1 and t_2.

In the case of balanced binary parse trees for instance, that means that for a tree $t^{(h)}$ of height h and a tree $t^{(h+1)}$ of height $h+1$, we have

$$s_{\text{IF}}(t^{(h+1)}) = \left(s_{\text{IF}}(t^{(h)}) + 1\right)^2 .$$

That makes clear why in DOP1, where fragments are assigned weights proportionally to their numbers of occurrence in the fragment corpus, the full parse trees with the greatest height, containing a great deal more fragments than parse trees of lesser height, are unjustifiably favored.

A similar calculation shows that the fragment multiset of a given full parse tree contains considerably more *fragments* of a certain height than fragments of any smaller height. This bias towards large *fragments* was addressed in two attempts of redefining the weight assignment function. The following subsection gives a brief overview.

2.2.5 Other Existing DOP Estimators

Bonnema's Estimator

Bonnema *et al.* [Bonnema *et al.*, 1999, Bonnema, 2003] propose a DOP estimation method that tackles the problem of DOP1's bias towards large fragments. We will omit its theoretical motivation here and just state the formula for the estimator directly. Let $N(f)$ denote the number of non-root nonterminal nodes of a fragment f. Bonnema's DOP estimator assigns to each fragment f from the training corpus TC the weight

$$\pi(f) = 2^{-N(f)} \frac{\boldsymbol{Count}_{\mathbf{Frag}_{\text{TC}}}(f)}{\boldsymbol{Count}_{\mathbf{Frag}_{\text{TC}}}(\{f' \in \mathbf{Frag}_{\text{TC}} \mid \text{root}(f') = \text{root}(f)\})} .$$

This way, weight mass is discounted from large fragments and distributed over smaller ones.

Is Bonnema's estimator consistent? In [Bonnema, 2003], a proof is given from which easily follows that the estimator is consistent for PCFG distributions: Given a probability distribution P_G over **Parses** resulting from a PCFG G, the sequence of DOP probability distributions estimated from growing training corpora sampled according to P_G converges to P_G. This is nice, but not nice enough: the simple example treebank from Section 2.2.3, for which DOP1 was shown to be inconsistent, also causes Bonnema's estimator to fail, as can easily be calculated.

Furthermore, [Sima'an & Buratto, 2003] report very disappointing results when testing the estimator on the OVIS corpus.

The DOP Maximum-Likelihood Estimator

The *DOP maximum-likelihood estimator* DOP_{ML} is the estimator defined by the following weight assignment to fragments f extracted from the training corpus TC:

$$\pi(f) = \begin{cases} \boldsymbol{rf}_{\text{TC}}(f) & \text{if } f \text{ is a full-parse tree in TC} \\ 0 & \text{otherwise} \end{cases}$$

The resulting DOP probability distribution is simply the relative frequency distribution of the full parse trees in the training corpus.

We will come back to this estimator in Section 2.3. As we will see in Section 3.3, DOP_{ML} is consistent.

Back-off DOP

In the method of *back-off parameter estimation* for the DOP model [Sima'an & Buratto, 2003], derivations are hierarchically structured within a so-called *back-off graph*. The aim is to account for the probabilistic dependencies of overlapping fragments in a principled manner (in contrast to DOP1, where simply independence is assumed in the generative model of a derivation process).

In order to define the back-off graph, the notion of a DOP derivation is slightly generalized to also allow the derivation of partial parses—*i.e.*, fragments—by a successive application of substitution steps. Further, the attention is restricted to length-two derivations, *i.e.*, pairs of fragments. A derivation $d = \langle f_1, f_2 \rangle$ is called a *back-off* of a fragment f if $f = f_1 \circ f_2$.

In [Sima'an & Buratto, 2003], length-two derivations form the nodes of the back-off graph. For the sake of presentation, we define the notion of a back-off graph slightly differently: A *back-off graph* is a directed graph whose nodes are the fragments from the training corpus and whose arrows point from fragment f to fragment g iff g participates in a back-off of f, *i.e.*, iff $f = g \circ h$ or $f = h \circ g$ for some fragment h. Note that this graph is acyclic since a fragment g participating in a back-off of a fragment f is always smaller in the number of nodes than f.

The back-off graph induces a hierarchy between copies of fragments—the *back-off hierarchy*—as follows: The first layer consists of the sources of the back-off graph, *i.e.*, the fragments that do not participate in any back-offs of fragments. These turn out to be the full parse trees of the training corpus. The n-th layer contains all fragments f for which there is a path of length $n-1$ from a source in the back-off graph to f. In other words, the n-th layer consists of the fragments that participate in back-offs of fragments from layer $n-1$. Note that the same fragment can occur in different layers! Since the back-off graph is acyclic and finite, the induced back-off hierarchy has always a finite number of layers.

The estimation procedure for Back-off DOP now operates iteratively in n steps, where n is the number of layers of the back-off hierarchy. In essence, this

procedure stepwisely transfers weight mass from one layer to the next. Intuitively, this amounts to the transfer of probability mass from preferred derivations to back-off derivations of trees in the training corpus.

The particular redistribution formula is an adaption of the Katz back-off method known from n-gram smoothing. However, as opposed to Katz back-off, where different estimators are interpolated during the testing process, Back-off DOP interpolates different *weight assignments* to fragments during the *training* process.

The original Katz smoothing formula assigns new probabilities to n-gram events based on up to n given probability distributions (usually the relative frequency distributions) over the set of all n-grams, the set of all $(n-1)$-grams, and so forth. In Back-off DOP, these n *given* distributions should correspond to n *given* distributions over fragments—one for each of the n layers in the back-off hierarchy. This confronts Back-off DOP with an inherently difficult problem: What *given* distributions over fragments should be used? What *meaning* does a probability distribution over fragments have for DOP in the first place? DOP assigns only probabilities to full parse trees, and this only indirectly: They result from weight assignments to fragments. Since the weight assignments to fragments of the same root form a probability distribution, [Sima'an & Buratto, 2003] choose to apply the back-off formula separately for each nonterminal N to fragments of root N. As *given* distributions, they use the weight assignments given by DOP1 and Bonnema-DOP, leading to two different variants of Back-off DOP.

We believe that

1. the back-off formula specified in [Sima'an & Buratto, 2003] does meet the intuition of assigning higher DOP probabilities to derivations from 'preferred' fragments in top (*i.e.*, low-numbered) layers of the back-off hierarchy and lower ones to back-off derivations from fragments in higher-numbered layers[6] and

2. applying back-off smoothing to weight assignments rather than DOP probabilities does not take into account an event's probability, as required in the Katz back-off formula, appropriately.

The first point is unproblematic. A slight adaption of the formula should suffice. As to the second point, how can the notion of a DOP probability, so far only defined for full parse trees, be generalized to fragments in a sensible way? In [Nguyen, 2004], the notions of a *DOP derivation* and its probability are generalized to fragments. For fragments that are full parse trees, her notion of a DOP probability coincides with the original notion.

[6]Note, however, that this was intended in that version of Back-off DOP: The aim was to countervail DOP1's bias towards large fragments by redistributing weight mass towards smaller fragments.

Unfortunately, proceeding in this way (using DOP probabilities instead of weights in the back-off formula) would lead to a system of nonlinear equations, one for each fragment occurrence in the back-off hierarchy. Sensible simplifying assumptions would be necessary.

An alternative solution could be to train n different estimators for each of the n layers in the back-off hierarchy and interpolate these estimators during the testing process, as done in the original Katz back-off estimation process. However, as Sima'an remarks (personal communication), such an estimation method would lose out considerably on DOP's original spirit of involving all kinds of fragments in the derivation process since fragments of different layers are now strictly separated into different estimation modules and could not recombine with each other.

According to Sima'an (personal communication), it should not be too difficult to devise the back-off formula in such a way that the resulting estimator becomes consistent. In essence, the fragments of the first layer in the back-off hierarchy would have to become increasingly dominant (in terms of assigned weight mass) as the corpus gets larger and more homogenous. Even the version of Back-off DOP presented in [Sima'an & Buratto, 2003] achieves already very promising empirical results. However, the inefficiency problem of DOP1 is inherited by Back-off DOP: Since all fragments are extracted from the training corpus, parsing time does not improve over DOP1.

2.3 Probability Models and The Principle of Maximum-Likelihood Estimation

If we look at the training process of a parsing system from the viewpoint of estimation theory, we can regard that process as the choice of a probability distribution over **Parses** out of a certain set $\mathcal{M}$ (called: *probability model*) of acceptable probability distributions. Often, for the choice of the appropriate probability distribution, the *Maximum-Likelihood Criterion* is used, according to which the probability distribution that maximizes the joint probability of the full parse trees in the training corpus, the so-called *Maximum-Likelihood Estimate (MLE)*, is chosen. If $\mathcal{M}$ is *unrestricted*, that is, if it contains all possible probability distributions over **Parses**, then the MLE always exists, is furthermore unique, and is identical to the *relative frequency estimate (RFE)*[7]

$$P_{\mathrm{rf}} : \mathbf{Parses} \rightarrow [0,1], \text{ where } P_{\mathrm{rf}}(t) := \boldsymbol{rf}_{\mathrm{TC}}(t)\,.$$

Remember here that $\boldsymbol{rf}_{\mathrm{TC}}(t) = \frac{\boldsymbol{Count}_{\mathrm{TC}}(t)}{|\mathrm{TC}|}$ denotes t's relative frequency in the training corpus TC.

[7] A proof of this can for example be found in [Ney *et al.*, 1997], Subsection 2.4, or [Prescher, 2003], Section 2.

Given a training corpus TC, the *DOP probability model induced by* TC, denoted $\mathcal{M}_{\text{DOP}}(\text{TC})$, is the set of all DOP-probability distributions (over **Parses**) resulting from weight assignments to the set of fragments in the fragment corpus $\mathbf{Frag}_{\text{TC}}$. This model plays a role not only for the problem of finding the most suitable weight assignment to the fragments from TC, but also for considering formal properties of DOP, such as bias and consistency, which will be the aim of Chapter 3.

For now, let us come back to the problem of maximum-likelihood estimation: The *DOP maximum-likelihood estimator* DOP_{ML} is usually referred to as the DOP procedure that assigns weights to the fragments extracted from the training corpus in such a way that the resulting joint DOP-probability of the training corpus is maximized [Bonnema *et al.*, 1999, Sima'an & Buratto, 2003]. DOP_{ML} thus yields the maximum-likelihood estimate w.r.t. $\mathcal{M}_{\text{DOP}}(\text{TC})$. Although the probability model is now restricted, the RFE belongs to it, and hence the MLE is still identical to the RFE. Why the RFE is a member of the probability model can easily be seen by considering the weight assignment allocating to each full parse tree its relative frequency in the training corpus and zero to all proper fragments, *i.e.*, fragments that are not full parse trees. All derivations remaining possible under this weight assignment are the unique length-1 derivations of the corpus trees themselves, having as their DOP probability the weight of the respective full parse, which is equal to its relative frequency.

What we also learn from this is that relative frequency estimation for DOP (and thus the DOP maximum-likelihood estimator) is of no practical use since it only assigns nonzero weights to the fragments that occur as full parse trees in the training data. This is not what we want. Natural language data is *sparse*, that is, many full parse trees will occur only a few times or not at all in the training data. Therefore, the RFE is unsatisfactory for statistical parsing. Not only does it assign probability 0 to all parses that do not occur in the training data. It can also only give rough estimates for rarely occurring full parse trees. There are different ways out of this common problem of *overfitting* in statistical NLP: One can either restrain the probability model $\mathcal{M}$ of acceptable probability distributions from which the MLE is chosen, or modify the model (*e.g.* by *pruning* it [Sima'an, 1999]), or adjust the relative frequency estimate by *discounting* probability mass from frequently occurring events (in our case: full parse trees) and distributing that mass over the unknown and rarely occurring events.[8] The technique of *held-out estimation*, which we will encounter in the following section, belongs to the class of discounting methods. *Restraining* the model (in the case of DOP) can for instance be achieved by only allowing fragments up to a certain depth to have nonzero weights [Bod, 2000, Bod, 2001] or by imposing conditions

[8] Actually, some discounting methods are just instances of MLE for a restrained probability model. In [Ney *et al.*, 1997] for instance, Ney derives the well-known formula for Good-Turing discounting as a solution of a maximum-likelihood problem.

on the weight assignment functions, as done in Back-off DOP (cf. Subsection 2.2.5).

Where will our estimation method DOP* fit in? Basically, DOP* restrains the model by only extracting fragments from a part of the training corpus. At the same time, however, some probability mass will be reserved and distributed over fragments from the rest of the corpus.

2.4 Held-Out Estimation

Held-out estimation is a method used in n-gram based language modelling. For comprehension of the rest of this thesis, the hurried reader only needs to know its core idea of reserving a part of the training corpus (the *held-out* part) for some preliminary testing on how representative the training data is and can now lightheartedly jump to the next chapter.

A simple example application for n-gram modelling (or equivalently: $(n-1)^{\text{th}}$ order Markov modelling) is the task of predicting the next word w in a text in a natural language by considering the $n-1$ words $w_1, \ldots, w_{n-1}$ that preceded w in the text. For this purpose, a probability distribution over n-tuples of words (called: *n-grams*) is estimated. This can be done by maximum-likelihood estimation, *i.e.*, by assigning each n-gram its relative frequency of occurrence in the training corpus TC. We have already encountered a related version of this estimation method for the case $n = 1$ (unigrams) at the beginning of this chapter.

The probability for a word w to occur after a word sequence $w_1 \ldots w_{n-1}$ can now be calculated as

$$P(w|w_1 \ldots w_{n-1}) = \frac{P(w_1 \ldots w_{n-1} w)}{P(w_1 \ldots w_{n-1})} = \frac{\boldsymbol{rf}_{\text{TC}}(w_1 \ldots w_{n-1} w)}{\sum_{w'} \boldsymbol{rf}_{\text{TC}}(w_1 \ldots w_{n-1} w')}$$

As we have seen in the previous section, relative frequency estimation is incapable of dealing with unknown events (here, n-grams). Therefore, in held-out estimation, some probability mass is discounted in a certain way from all *known* n-grams (*i.e.*, n-grams that occurred during training) and evenly distributed over the unknown ones. Here, *unknown* n-grams are meant to be n-grams from the set **NGrams** which are not known, where **NGrams** is defined dependent on the training data as the set of n-tuples of the set of words from the training corpus. The set **NGrams** is therefore always finite, whence it is possible to evenly distribute the discounted probability mass over the unknown n-grams.

How should we determine the total amount of probability mass by which the RFE n-gram probabilities are discounted? In held-out estimation, the training corpus TC is split into *actual training corpus* ATC and *held-out corpus* HC. The actual training corpus is used to obtain the RFE probability distribution for the

n-grams as explained above. Then the held-out corpus is considered in order to estimate how often n-grams that occurred r times actually happen to occur during testing. This yields an estimate of the expected relative frequency of an n-gram during testing and thus of its actual probability.

Let $r = \boldsymbol{Count}_{\text{ATC}}(t)$ be the number of occurrences of a certain n-gram t in ATC. Then we define

$$T_r := \sum_{t' :\, \boldsymbol{Count}_{\text{ATC}}(t') = r} \boldsymbol{Count}_{\text{HC}}(t')$$

to be the total number of times that n-grams occurring r times in the actual training data occur in the held-out data. Dividing T_r by the number N_r of distinct n-grams appearing exactly r times in the actual training data yields the average frequency that such n-grams have in the held-out data. This value is called the *held-out discount* $\widetilde{r}_{\text{ho}}$ of r:

$$\widetilde{r}_{\text{ho}} := \frac{T_r}{N_r}, \text{ where } N_r := |\{t' \in \mathbf{NGrams} \mid \boldsymbol{Count}_{\text{ATC}}(t') = r\}|$$

Note that for unknown n-grams t (*i.e.*, $r = 0$), N_0 is the number of distinct n-grams not occurring in ATC, *i.e.*, the number of distinct unknown n-grams. Note also that although dubbed 'discount', $\widetilde{r}_{\text{ho}}$ can actually be greater than r even for $r \geq 1$.

The held-out discount still depends on the size of the held-out data. Dividing by that size yields a held-out estimate of the relative frequency with which an n-gram occurs in actual testing data, and thus an improved estimate of an n-gram's probability as compared to relative frequency estimation:

$$P_{\text{ho}}(t) := \frac{\widetilde{r}_{\text{ho}}}{|\text{HC}|}, \text{ where } r = \boldsymbol{Count}_{\text{ATC}}(t) \tag{2.2}$$

It is easy to check that P_{ho} is a probability distribution over **NGrams**.

Unfortunately, held-out estimation is not straight-forwardly applicable to DOP, the reason being that it is ultimately based on relative frequency estimation. Relative frequency estimation for DOP is of no use since it only assigns probability mass to full parse trees directly occurring in the training data. Held-out estimation applied to DOP would just distribute the reserved probability mass evenly over all 'unknown' parse trees (where the definition of 'unknown' would here have to be adjusted in order to make the set of all unknown parses finite), treating them all completely equally. The actual strength of DOP, lying in the way it combines evidence for full parse trees found in the training data in order to predict the existence of similar full parse trees which have not occurred during training, would be completely ignored by held-out estimation.

Nevertheless, the idea of dividing the training corpus into two parts, using the first one for training, and then adjusting certain parameters (in our case, the

fragment weights) by testing on the second one, seems appealing and will actually be utilized by DOP*.

Chapter 3

Considerations about Bias and Consistency

In this chapter, we will introduce the notion of an estimator and the properties of bias and consistency. Subsequently, we examine how bias and consistency apply to DOP. We will show that every reasonable DOP estimator must be biased (Section 3.2), give a proof that the DOP maximum-likelihood estimator is consistent in preparation for the consistency proof for DOP* (Section 3.3), and informally ascertain necessary conditions a consistent DOP estimator must fulfill (Section 3.4).

3.1 Basic Notions

In the following, we will establish the notion of an estimator and some of its properties. An introductory treatment of estimation theory is for instance given in [DeGroot & Schervish, 2002], Chapter 6; [Krenn & Samuelsson, 1997], Section 1.7; or [Siegrist, 2004]. However, in statistical parsing, we are interested in the estimation of whole probability distributions, not merely real-valued parameters or parameter vectors in $\mathbb{R}^n$, as in standard estimation theory. Therefore, the standard definitions have to be slightly adapted.

3.1.1 Estimation

Informally speaking, the training procedure in statistical parsing results in a probability distribution, according to which the parser can determine the preferred parse for an input sentence. We will call this procedure an *estimator* and the training data its *observations*. Intuitively, an estimator thus tries to approximate the probability distribution P that underlies the observations made. In the case

of DOP, the estimator is the procedure that assigns weights to fragments and thereby probabilities to full parse trees dependent on the training corpus.

Usually, assumptions are made on the kind of probability distributions that can underly the training data. This translates to *fixing a model* in the jargon of estimation theory: The training corpus is assumed to have been sampled from a probability distribution in a fixed model $\mathcal{M}$ (cf. Section 2.3).

Assume thus we are given a model $\mathcal{M}$ of probability distributions over the sample space **Parses** and a probability distribution $P \in \mathcal{M}$, according to which independent samples, *observations*, are drawn. An estimator tries to predict P from the *observation sequence*—a sequence of random samples from **Parses** according to the probability distribution P, in the case of DOP the training corpus. The actual definition of an estimator is independent of $\mathcal{M}$:

Let $\mathcal{M}_0$ denote the unrestricted probability model over **Parses** (cf. Section 2.3). An *estimator* $\varphi : \mathbf{Parses}^* \to \mathcal{M}_0$ is a function that assigns a probability distribution (the *estimate*) to a finite sequence of samples from **Parses**.[1]

The model $\mathcal{M}$ will become important when we consider *properties* of φ, such as bias and consistency. The model-independent definition of an estimator also enables us to consider properties of one estimator w.r.t. different underlying models, which we will actually do later.

In the following, we will sometimes denote the estimate $\varphi(s)$ for an observation sequence $s = \langle t_1, \ldots, t_n \rangle \in \mathbf{Parses}^n$ as φ_s to stress the fact that it is a probability distribution.

3.1.2 Bias, Loss Function, and Consistency

In the following definitions, we adhere to [Johnson, 2002], where the inconsistency of DOP1 was first proved, using slightly simplified notation.

Let $X = \langle X_1, \ldots, X_n \rangle$ be a sequence of n independent random variables distributed according to a probability distribution P in the model $\mathcal{M}$. Then the estimate $\varphi(X)$ is a random variable as well, ranging over the probability distributions in $\mathcal{M}_0$. It is easy to see that the expected value of the probability distribution assigned by φ,

$$\mathrm{E}_P\left[\varphi(X)\right] = \sum_{\langle t_1,\ldots,t_n\rangle \in \mathbf{Parses}^n} P(t_1)\cdots P(t_n)\varphi(t_1,\ldots,t_n)\ ,$$

is also a probability distribution over **Parses**.

[1] Recall that for a set S, $S^* := \bigcup_{i\in\mathbb{N}} S^i$.

Bias

Based on the expected value of $\varphi(X)$, we define the estimator φ to be *biased* for some probability distribution P over **Parses** 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,

$$\mathrm{E}_P\left[\varphi(X)\right] \neq P$$

holds. We call φ *biased w.r.t.* $\mathcal{M}$ if it is biased for some $P \in \mathcal{M}$.

Loss

A *loss function* $\mathcal{L}$ is a mapping from $\mathcal{M}_0^2$ to the nonnegative reals. The value $\mathcal{L}(P, \varphi(t_1, \ldots, t_n))$ expresses the loss incurred by the error made in the estimate φ_{TC} of P from the sample sequence $\text{TC} = \langle t_1, \ldots, t_n \rangle$.

The expected loss for an estimation of P from a sequence of observations of length n,

$$\mathrm{E}_P[\mathcal{L}(P, \varphi(X_1, \ldots, X_n))]$$

is called the *risk* of φ at P for sample size n.

Consistency

The estimator φ is called *consistent w.r.t.* $\mathcal{M}$ if for each probability distribution $P \in \mathcal{M}$, the risk of φ at P approaches zero when the sample size goes to infinity, *i.e.*, if we have

$$\lim_{n\to\infty} \mathrm{E}_P\left[\mathcal{L}(P, \varphi(X_1, \ldots, X_n))\right] = 0$$

for all $P \in \mathcal{M}$.

The question remaining is: What loss function should we choose in order to obtain a sensible definition of consistency? We follow [Johnson, 2002] by defining

$$\mathcal{L}(P, \varphi(t_1, \ldots, t_n)) := \sum_{t \in \mathbf{Parses}} P(t)\left[P(t) - \varphi(t_1, \ldots, t_n)(t)\right]^2 \ .$$

Note that the function value is always in $[0, 1]$ since

$$0 \leq \left[P(t) - \varphi(t_1, \ldots, t_n)(t)\right]^2 \leq 1 \ .$$

Writing out the formula for the expected loss, φ is thus consistent w.r.t. $\mathcal{M}$ iff

$$\lim_{n\to\infty} \sum_{\text{TC} \in \mathbf{Parses}^n} P(\text{TC}) \sum_{t \in \mathbf{Parses}} P(t)\left[P(t) - \varphi_{\text{TC}}(t)\right]^2 = 0 \ \text{ for all } P \in \mathcal{M},$$

where $P(\text{TC}) = P(t_1) \cdots P(t_n)$ is the probability of the sequence $\text{TC} = \langle t_1, \ldots, t_n \rangle$ of independent samples from **Parses** drawn according to P.

Strong consistency

In the literature, consistency is often defined more directly 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 $\mathcal{M}$ and φ be given as specified above. Then φ is called *strongly consistent w.r.t.* $\mathcal{M}$ if for each $P \in \mathcal{M}$ and each real number $\varepsilon > 0$, we have

$$\lim_{n\to\infty} \sup_{t\in\mathbf{Parses}} \sum_{\substack{\text{TC}\in\mathbf{Parses}^n: \\ |\varphi_{\text{TC}}(t)-P(t)|\geq\varepsilon}} P(\text{TC}) = 0 \; .$$

3.1.3 Strong Consistency Implies Consistency

As our denotation suggests, strong consistency implies consistency:

Theorem 3.1.1 *Let $\mathcal{M}$ be a probability model over* **Parses** *and $\varphi : \mathbf{Parses}^* \to \mathcal{M}_0$ an estimator.*

If φ is strongly consistent w.r.t. $\mathcal{M}$ then φ is also consistent w.r.t. $\mathcal{M}$.

Proof: Assume that φ is strongly consistent w.r.t. $\mathcal{M}$. Further, let P be a probability distribution in $\mathcal{M}$. We have to show:

$$\lim_{n\to\infty} \sum_{\text{TC}\in\mathbf{Parses}^n} P(\text{TC}) \sum_{t\in\mathbf{Parses}} P(t)\,[P(t) - \varphi_{\text{TC}}(t)]^2 = 0$$

Assume thus, we are given $\varepsilon' > 0$. Now define $\varepsilon := \sqrt{\varepsilon'/2}$ and $q := \varepsilon'/2$. Since φ is strongly consistent w.r.t. $\mathcal{M}$, there is an $N \in \mathbb{N}$ such that for all $n \in \mathbb{N}$ with $n \geq N$, we have

$$\sup_{t\in\mathbf{Parses}} \sum_{\substack{\text{TC}\in\mathbf{Parses}^n: \\ |\varphi_{\text{TC}}(t)-P(t)|\geq\varepsilon}} P(\text{TC}) \leq q$$

and hence

$$\sum_{\substack{\text{TC}\in\mathbf{Parses}^n: \\ |\varphi_{\text{TC}}(t)-P(t)|\geq\varepsilon}} P(\text{TC}) \leq q \quad (\text{for all } t \in \mathbf{Parses}) \tag{3.1}$$

and hence

$$
\begin{aligned}
&\sum_{\text{TC}\in\mathbf{Parses}^n} P(\text{TC}) \sum_{t\in\mathbf{Parses}} P(t)\,[P(t)-\varphi_{\text{TC}}(t)]^2 \\
&\quad= \sum_{t\in\mathbf{Parses}} P(t) \sum_{\text{TC}\in\mathbf{Parses}^n} P(\text{TC})\,[P(t)-\varphi_{\text{TC}}(t)]^2 \\
&\quad\le \sup_{t\in\mathbf{Parses}} \sum_{\text{TC}\in\mathbf{Parses}^n} P(\text{TC})\,[P(t)-\varphi_{\text{TC}}(t)]^2 \\
&\quad\le \sup_{t\in\mathbf{Parses}} \Bigg[\underbrace{\sum_{\substack{\text{TC}\in\mathbf{Parses}^n:\\ |P(t)-\varphi_{\text{TC}}(t)|<\varepsilon}} P(\text{TC}) \underbrace{[P(t)-\varphi_{\text{TC}}(t)]^2}_{<\varepsilon^2}}_{<\varepsilon^2} \\
&\qquad\qquad + \underbrace{\sum_{\substack{\text{TC}\in\mathbf{Parses}^n:\\ |P(t)-\varphi_{\text{TC}}(t)|\ge\varepsilon}} P(\text{TC}) \underbrace{[P(t)-\varphi_{\text{TC}}(t)]^2}_{\le 1}}_{\le q \text{ by (3.1)}} \Bigg] \\
&\quad\le \varepsilon^2 + q = \varepsilon' \,.
\end{aligned}
$$

QED

We conjecture that the other direction also holds, *i.e.*, that the properties of consistency and strong consistency are actually equivalent. Be that as it may, in the consistency proofs given in this thesis, we will show *strong* consistency, and thereby also consistency, of the respective estimators.

3.2 A Short Word on Bias

In estimation theory, *being unbiased* is often considered a quality criterion for an estimator (see, *e.g.*, [Krenn & Samuelsson, 1997], Subsection 1.7.2). However, as illustrated for instance in [DeGroot & Schervish, 2002], Section 7.7, there are estimation problems in which the aim of unbiased estimation is of limited, if not counterproductive, utility. How do matters stand with DOP? In [Johnson, 2002], where DOP1 is shown to be biased and inconsistent, it is conjectured that "undoubtedly there are other estimation procedures for DOP models which are unbiased and consistent." Certainly, the standard DOP maximum-likelihood estimator is, but as we have seen in Sections 2.3 and 2.4, that estimator in its pure form is of no use to DOP because it assigns probability zero to all full parse trees that do not occur directly in the training data. Could any *reasonable* DOP estimator (that is, an estimator that doesn't completely overfit the training data) be unbiased? As we will see in this section, bias of the DOP estimator is a necessary (and desired) by-product of DOP's basic conception of assigning nonzero

probabilities to full parse trees not occurring in the training data. We start with a theorem that gives a necessary condition for a (general) estimator to be biased for a particular probability distribution.

Theorem 3.2.1 *Let $\varphi : \mathbf{Parses}^* \to \mathcal{M}_0$ an estimator for which there is a training corpus $\text{TC} = \langle t_1, \ldots, t_n \rangle \in \mathbf{Parses}^n$ and a full parse tree t_0 outside the corpus (*i.e.*, $t_0 \neq t_i$ $(i = 1, \ldots, n)$) such that*

$$\varphi_{\text{TC}}(t_0) > 0 \; .$$

Then φ is biased for each probability distribution P over **Parses** *that assigns a positive probability to* TC *but a zero-probability to t_0,* i.e.*, for which $P(t_1) \cdots P(t_n) > 0$ and $P(t_0) = 0$.*

Proof: Let φ and $\text{TC} = \langle t_1, \ldots, t_n \rangle$ be given as specified above and assume φ is unbiased for some probability distribution P with $P(t_1) \cdots P(t_n) > 0$ and $P(t_0) = 0$. This means that

$$\mathrm{E}_P(\varphi(X_1, \ldots, X_n)) = \sum_{\substack{\langle \omega_1, \ldots, \omega_n \rangle \\ \in \mathbf{Parses}^n}} P(\omega_1) \cdots P(\omega_n) \varphi(\omega_1, \ldots, \omega_n) = P. \tag{3.2}$$

Thus, we have

$$\sum_{\substack{\omega \in \mathbf{Parses}: \\ P(\omega) \neq 0}} \sum_{\substack{\langle \omega_1, \ldots, \omega_n \rangle \\ \in \mathbf{Parses}^n}} P(\omega_1) \cdots P(\omega_n) \varphi(\omega_1, \ldots, \omega_n)(\omega) = \sum_{\substack{\omega \in \mathbf{Parses}: \\ P(\omega) \neq 0}} P(\omega). \tag{3.3}$$

Since $\sum_{\omega \in \mathbf{Parses}:\, P(\omega) \neq 0} P(\omega) = 1$, we obtain from (3.3):

$$\sum_{\substack{\omega \in \mathbf{Parses}: \\ P(\omega) \neq 0}} \sum_{\substack{\langle \omega_1, \ldots, \omega_n \rangle \\ \in \mathbf{Parses}^n}} P(\omega_1) \cdots P(\omega_n) [\varphi(\omega_1, \ldots, \omega_n)](\omega) = 1, \tag{3.4}$$

i.e.,

$$\sum_{\substack{\langle \omega_1, \ldots, \omega_n \rangle \\ \in \mathbf{Parses}^n}} P(\omega_1) \cdots P(\omega_n) \sum_{\substack{\omega \in \mathbf{Parses}: \\ P(\omega) \neq 0}} [\varphi(\omega_1, \ldots, \omega_n)](\omega) = 1. \tag{3.5}$$

Since

$$\sum_{\substack{\langle \omega_1, \ldots, \omega_n \rangle \\ \in \mathbf{Parses}^n}} P(\omega_1) \cdots P(\omega_n) = 1$$

and

$$\sum_{\substack{\omega \in \mathbf{Parses}: \\ P(\omega) \neq 0}} [\varphi(\omega_1, \ldots, \omega_n)](\omega) \leq 1,$$

Equation (3.5) can only be valid if $\sum_{\{\omega \in \mathbf{Parses}:\, P(\omega) \neq 0\}} [\varphi(\omega_1, \ldots, \omega_n)](\omega) = 1$ for all $\omega_1, \ldots, \omega_n \in \mathbf{Parses}$ such that $P(\omega_1) \cdots P(\omega_n) > 0$. But this means $[\varphi(\omega_1, \ldots, \omega_n)](\omega) = 0$ for all $\omega, \omega_1, \ldots, \omega_n \in \mathbf{Parses}$ with $P(\omega) = 0$ and $P(\omega_1) \cdots P(\omega_n) > 0$. Thus, $[\varphi(t_1, \ldots, t_n)](t_0) = 0$, which is a contradiction. QED

Now we apply the theorem to DOP. The following corollary states that, given a training corpus TC and a DOP estimator that is unbiased w.r.t. TC's induced DOP probability model[2] $\mathcal{M}_{\text{DOP}}(\text{TC})$, the estimator is bound to completely overfit the training corpus by assigning zero-probabilities to all full parse trees outside the corpus.

Corollary 3.2.2 *Let there be a training corpus* $\text{TC} \in \mathbf{Parses}^n$ *and a DOP estimator* $\varphi : \mathbf{Parses}^* \rightarrow \mathcal{M}_0$ *that is unbiased w.r.t.* $\mathcal{M}_{\text{DOP}}(\text{TC})$. *Then* $\varphi_{\text{TC}}(t) = 0$ *for all* $t \in \mathbf{Parses}$ *with* $t \notin \text{TC}$.

Proof: Assume indirectly that $\varphi_{\text{TC}}(t_0) > 0$ for some full parse tree t_0 that is not in TC. As shown in Section 2.3, the relative frequency estimate

$$P_{\boldsymbol{rf}_{\text{TC}}} : \mathbf{Parses} \rightarrow [0,1], \text{ where} \tag{3.6}$$

$$P_{\boldsymbol{rf}_{\text{TC}}}(t) := \boldsymbol{rf}_{\text{TC}}(t) \tag{3.7}$$

is an instance of $\mathcal{M}_{\text{DOP}}(\text{TC})$. Since $\boldsymbol{rf}_{\text{TC}}(t) > 0$ for all $t \in \text{TC}$ and $\boldsymbol{rf}_{\text{TC}}(t_0) = 0$, it follows from Theorem 3.2.1 that φ is biased for $P_{\boldsymbol{rf}_{\text{TC}}}$. Thus φ is biased w.r.t. $\mathcal{M}_{\text{DOP}}(\text{TC})$. QED

It might be of interest to apply Theorem 3.2.1 to other estimators in statistical NLP. As pointed out in [Prescher *et al.*, 2004], the theorem is *not* of relevance to probabilistic context free grammars (PCFGs) since the PCFG model $\mathcal{M}_{\text{PCFG}}(\text{TC})$ induced by a training corpus TC does not contain a probability distribution that assigns positive probabilities to the trees in TC and zero to an outside tree.

3.3 The DOP Maximum-Likelihood Estimator Is Consistent

Although it is generally accepted that the DOP maximum-likelihood estimator DOP_{ML}, introduced in Subsection 2.3, is consistent, no such proof exists in the literature so far. Remember that DOP_{ML} assigns each full parse tree its relative frequency in the training corpus and is thus identical to the relative frequency estimator. Relative frequency estimation for DOP differs from standard textbook RFE in that a DOP estimator does not estimate one single real-valued parameter or a parameter vector in $\mathbb{R}^n$ of a probability distribution, but rather the probability distribution *itself*. Therefore, the results for standard RFE cannot be utilized.

In the following, we will give a proof that DOP_{ML} is consistent. We will actually show that the estimator is *strongly* consistent—not only w.r.t. the probability model $\mathcal{M}_{\text{DOP}}(\text{TC})$ induced by a given training corpus TC (cf. Section 2.3)

[2] Confer Section 2.3 for a Definition of $\mathcal{M}_{\text{DOP}}(\text{TC})$.

but even w.r.t. the unrestricted model $\mathcal{M}_0$ of all probability distributions over **Parses**. The core part of the proof can be employed in the consistency proof of DOP* given in the next chapter and is therefore stated separately as a lemma.

Lemma 3.3.1 *Let P be a probability distribution over* **Parses**. *Then for each full parse tree $t \in$ **Parses**, natural number $n \in \mathbb{N}$ and real value $\varepsilon > 0$, it holds that*

$$\sum_{\substack{\text{TC} \in \mathbf{Parses}^n: \\ |\boldsymbol{rf}_{\text{TC}}(t) - P(t)| \geq \varepsilon}} P(\text{TC}) \leq \frac{1}{4n\varepsilon^2} \, .$$

Proof: Let P, t, n and ε be defined as above. Chebyshev's inequality states that for any real-valued random variable X on $\mathbf{Parses}^n$ with expected value μ and variance σ^2 and any $\varepsilon' > 0$, we have

$$P(|X - \mu| \geq \varepsilon') \leq \frac{\sigma^2}{\varepsilon'^2} \, ,$$

i.e.,

$$\sum_{\substack{\text{TC} \in \mathbf{Parses}^n: \\ |X(\text{TC}) - \mu| \geq \varepsilon'}} P(\text{TC}) \leq \frac{\sigma^2}{\varepsilon'^2} \, .$$

The relative frequency $\boldsymbol{rf}_{\text{TC}}(t)$ of t in TC is a random variable on $\mathbf{Parses}^n$ with the expected value $\mu = p$ and the variance $\sigma^2 = p(1-p)/n$, where $p = P(t)$. Thus, applying Chebyshev's inequality yields

$$\sum_{\substack{\text{TC} \in \mathbf{Parses}^n: \\ |\boldsymbol{rf}_{\text{TC}}(t) - P(t)| \geq \varepsilon}} P(\text{TC}) \leq \frac{\overbrace{P(t)[1 - P(t)]}^{\leq 1/4}}{n\varepsilon^2} \leq \frac{1}{4n\varepsilon^2} \, .$$

QED

Now we are ready for the consistency theorem:

Theorem 3.3.2 DOP_{ML} *is strongly consistent w.r.t. the model $\mathcal{M}_0$ of all probability distributions over* **Parses**.

Proof: First note that the estimate $\text{DOP}_{\text{ML}}(\text{TC})$ is a probability distribution assigning each full parse tree $t \in$ **Parses** its relative frequency $\boldsymbol{rf}_{\text{TC}}(t)$. Now let P be a probability distribution over **Parses** and ε and q two positive real numbers. We will give an $N \in \mathbb{N}$ such that for each $n \in \mathbb{N}$ with $n \geq N$, we have

$$\sup_{t \in \mathbf{Parses}} \sum_{\substack{\text{TC} \in \mathbf{Parses}^n: \\ |\boldsymbol{rf}_{\text{TC}}(t) - P(t)| \geq \varepsilon}} P(\text{TC}) \leq q \, . \tag{3.8}$$

From that follows the claim.

Define N to be the smallest natural number such that $N \geq \frac{1}{4\varepsilon^2 q}$. Then Lemma 3.3.1 yields for all $n \geq N$ and $t \in \mathbf{Parses}$

$$\sum_{\substack{\text{TC} \in \mathbf{Parses}^n: \\ |rf_{\text{TC}}(t) - P(t)| \geq \varepsilon}} P(\text{TC}) \leq 1/\underbrace{(4n\varepsilon^2)}_{\geq 4\varepsilon^2 N} \leq q \; .$$

Equation (3.8) follows immediately. QED

3.4 Contemplation

Let us pause for a moment and ponder which kind of DOP estimators can actually achieve consistency. In the case of DOP, consistency means that when the size of the training corpus approaches infinity, the estimator's assignments of probabilities to full parse trees must converge to their relative frequencies in the training corpus. That is because the loss function approaches zero only if the DOP probability distribution assigned by the estimator approaches the 'true' distribution underlying the training corpus. But that distribution on its part is approached by the relative frequency distribution of the full parse trees in the training corpus when its size goes to infinity.

When the probability distribution assigned by the DOP estimator approaches the relative frequency distribution of the parse trees in the training corpus, this means that the probability assignments for all parse trees not found in the training corpus have to approach zero. This, however, should not happen too early (*i.e.*, when the sample size is not large enough), otherwise the estimator would *overfit* the data.

Chapter 4

The New Estimator DOP*

In the last section of the previous chapter, we have seen that the DOP probability distribution assigned by a *consistent* DOP estimator must approach the relative frequency distribution of the parse trees in the training corpus when the corpus size goes to infinity, and that the probability assignments for all parse trees not found in the training corpus therefore have to approach zero. We will now devise a DOP estimator DOP* which will have the property that as the training sample size approaches infinity, the probabilities assigned to derivations of length greater than one approach zero, while the weights (and thereby the probabilities) assigned to the full parse trees from the training corpus converge to their relative frequencies.

In Section 4.1, the DOP* estimation procedure is explained. A consistency proof for DOP* is given in Section 4.2. Further, we show that DOP* can achieve an exponential reduction in the number of fragments extracted from the training corpus w.r.t. DOP1 (Section 4.3) and argue that in contrast to DOP1, DOP* is not biased towards fragments of large full parse trees (Section 4.4).

4.1 The DOP* Estimation Procedure

As we have seen in Section 2.3, the standard method of maximum-likelihood estimation, according to which the joint probability of the full parse trees in the training corpus should be maximized, is not suitable for DOP. Given a training corpus TC, the MLE over $\mathcal{M}_{\text{DOP}}(\text{TC})$ assigns nonzero probability only to the full parse trees occurring directly in the training corpus, leading to an *overfitted* parser that can only reproduce the parses that occurred during training.

DOP* avoids overfitting by randomly splitting the training corpus into two parts: the *extraction corpus* EC and the *held-out corpus* HC. The exact method of division is not important in the following discussion, as long as both corpora's sizes approach infinity as $|\text{TC}|$ approaches infinity. (In practice, however, the

method of division is certainly of relevance, and we will come back to it in Chapter 5.) While fragments are extracted from the trees in EC, their weights are assigned such that the likelihood of the held-out corpus HC is maximized. It can happen that a full 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.

Crucially, to avoid Expectation-Maximization algorithms such as Inside-Outside [Baker, 1979] for approaching the MLE over HC[1], we will make the following simplifying assumption: maximizing the joint probability of the full parse trees in HC is equivalent to maximizing the joint probability of their *shortest derivations*. This assumption turns out handy for several reasons:

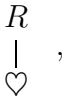
- It leads to a closed-form solution for the MLE, which is further computationally very inexpensive.
- The resulting estimator will only assign nonzero weights to a number of fragments that is linear in the number of depth-1 fragments (*i.e.*, PCFG rules) contained in HC, thereby resulting in an exponential 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 full parse tree than a longer one; thus the shortest derivation can be regarded as the preferred way of building up a full parse tree from fragments, and the longer derivations as provisional solutions (backoffs) that would have to be used if no shorter ones were available. Furthermore, there are empirical reasons to make the shortest derivation assumption: In [De Pauw, 1999, Bod, 2000, De Pauw, 2000] it is shown that DOP models that select the preferred parse of a test sentence using the shortest derivation criterion perform very well.

To ensure maximum coverage (*i.e.*, to have the estimator assign nonzero probabilities to as many parse trees as possible), the estimation procedure outlined above reserves a certain proportion of the weight mass for smoothing: In a second estimation step, also fragments that did not participate in any shortest derivation

[1]Inside-Outside is a hill-climbing algorithm for statistical parsing, which has been applied to DOP in [Bod, 2000]. Inside-Outside 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, doesn't [Charniak, 1993]) arrive at a global maximum of the likelihood function.

of HC-trees will be given some weight. For that purpose, an *imaginary fragment*



,

consisting only of the root R and its child terminal '♡', and denoted by $\heartsuit_R$, is introduced for each root R. The weight assigned to $\heartsuit_R$ stands for the weight mass to reserve and is chosen dependent on the relative frequency p_{unkn} of parse tree occurrences in HC that are not EC-derivable. The lower the value of p_{unkn}, the less weight mass is reserved. The smoothing algorithm then distributes for each nonterminal R the weight of the imaginary fragment $\heartsuit_R$ over all root-R fragments.

The DOP* estimation procedure thus consists of the following parts:

1. The core DOP* estimator, assigning weights $\beta(f)$ to fragments f from the extraction corpus. Thereby, for each nonterminal R, a certain amount $\beta(\heartsuit_R)$ of weight mass is reserved for the smoothing step.

2. The smoothing component, distributing the reserved weight mass $\beta(\heartsuit_R)$ over all fragments from the training corpus and assigning each fragment f the smoothing weight $\beta_{\text{smooth}}(f)$.

3. The final weight assignment π to fragments f, given by

$$\pi(f) := \beta(f) + \beta_{\text{smooth}}(f) \ .$$

In the following subsection, we will derive the formula for the preliminary weight assignment β as a solution to the maximum-likelihood problem indicated above. How the reserved weight masses $\beta(\heartsuit_R)$ are determined and distributed is explained in Subsection 4.1.2. Figure 4.1 gives a summary of the estimation algorithm.

4.1.1 Estimation of the β-weights

In order to assign the β-weights to the fragments in $\mathbf{Frag}_{\text{EC}}$, derivations of full parse trees in HC using the fragments in $\mathbf{Frag}_{\text{EC}}$ are considered. As in held-out estimation (cf. Section 2.4), the sum of the relative frequencies of the trees in HC that are not EC-derivable is taken as the estimate p_{unkn} of the probability that a tree will be unknown during testing. Depending on p_{unkn} in a certain way described in the next section, weight mass $\beta(\heartsuit_R)$ for each nonterminal R occurring as root in $\mathbf{Frag}_{\text{EC}}$ is assigned to the imaginary fragment $\heartsuit_R$ and thus reserved for the smoothing step of the estimation procedure. Note that the DOP probability distribution resulting from a weight assignment β assigns some probability mass

1. Split TC into EC and HC.

2. Extract the fragments from EC.

3. Determine $p_{\text{unkn}} = \dfrac{\boldsymbol{Count}_{\text{HC}}\left(\{t \in \textbf{Parses} \mid t \text{ is EC-derivable}\}\right)}{|\text{HC}|}$.

4. For each EC-derivable parse $t \in$ HC, determine its shortest derivation(s) $d^1(t), \ldots, d^{\#\text{shder}(t)}(t)$.

5. For all fragments $f_1, \ldots, f_N$ involved in shortest derivations of parses in HC, determine the parameters

$$r_k := \sum_{t \in \text{HC}} \frac{\boldsymbol{Count}_{\text{HC}}(t)}{\#\text{shder}(t)} \sum_{i=1}^{\#\text{shder}(t)} \boldsymbol{Count}_{d^i(t)}(f_k) \quad (k = 1, \ldots, N) \ .$$

6. For each nonterminal R in the whole training corpus TC, determine $\beta(\heartsuit_R)$, *e.g.* by setting

$$\beta(\heartsuit_R) := \begin{cases} p_{\text{unkn}} & \text{if } R \in \{R_1, \ldots, R_M\} \\ 1 & \text{otherwise} \end{cases} \ ,$$

   where $\{R_1, \ldots, R_M\}$ is the set of roots of the fragments $f_1, \ldots, f_N$.

7. For the fragments $f_1, \ldots, f_N$, set

$$\beta(f_j) := \left(1 - \beta\left(\heartsuit_{\text{root}(f_j)}\right)\right) \frac{r_j}{\sum\limits_{\substack{k \in \{1,\ldots,N\}: \\ \text{root}(f_k) = \text{root}(f_j)}} r_k} \ .$$

   For all other fragments $f \in \textbf{Frag}_{\text{TC}}$, set $\beta(f) := 0$.

8. For each fragment $f \in \textbf{Frag}_{\text{TC}}$, determine the final weight

$$\pi(f) := \beta(f) + \beta_{\text{smooth}}(f) \ ,$$

   where *e.g.*

$$\beta_{\text{smooth}}(f) := \frac{\beta\left(\heartsuit_{\text{root}(f)}\right) \ \boldsymbol{Count}_{\textbf{Frag}_{\text{TC}}}(f)}{\boldsymbol{Count}_{\textbf{Frag}_{\text{TC}}}\left(\{f' \mid \text{root}(f') = \text{root}(f)\}\right)} \ .$$

Figure 4.1: The DOP* estimation algorithm

to *imaginary trees*, that is, full parse trees that contain the terminal '♡' in their yields.

How do we assign the β-weights to the fragments in $\mathbf{Frag}_{\text{EC}}$ based on the sample sequence HC? In the following, we will set up a maximization problem for the weight assignment function β, in which we regard the reserved weight masses $\beta(\heartsuit_R)$ as constant (although not yet known). We derive β as the solution to the problem of maximizing the probability of HC$'$ w.r.t. the model $\mathcal{M}_{\text{DOP}}(\text{EC})$, where HC$'$ is the corpus obtained from HC by removing the trees that are not EC-derivable:

$$\arg\max_{\beta \in W} \prod_{\substack{t \in \text{HC}: \\ t \text{ is EC-derivable}}} [P_{\text{DOP}_\beta}(t)]^{\boldsymbol{Count}_{\text{HC}}(t)} \ , \tag{4.1}$$

where P_{DOP_β} denotes the DOP probability distribution over **Parses** resulting from the weight assignment β, and W is the set of all $\beta : \mathbf{FragSet}_{\text{EC}} \rightarrow [0,1]$ that fulfill the side conditions that for each nonterminal R in EC:

$$\sum_{f \in \mathbf{Frag}_{\text{EC}}: \, \text{root}(f)=R} \beta(f) + \beta(\heartsuit_R) = 1 \ . \tag{4.2}$$

Note that since the DOP-probability of a full parse tree is the sum of the DOP-probabilities of its derivations, the term to be maximized in problem (4.1) is a product of sums of products of weights. We now make the simplifying assumption indicated above that problem (4.1) remains equivalent when each DOP-probability $P_{\text{DOP}_\beta}(t)$ is replaced by the probability of the shortest derivation of t. (Note that it will turn out that the consistency of DOP* does **not** rely on this assumption!) If there are more than one shortest derivation for a parse (say n), we will assume that each of them derived $1/n$ of the occurrences of that parse, a fraction which needs not necessarily be a whole number. This leads us to the maximization problem

$$\arg\max_{\beta \in W} \prod_{t \in \text{HC}: \ t \text{ is EC-derivable}} \prod_{i=1}^{\#\text{shder}(t)} [P^i_{\text{sh}_\beta}(t)]^{\frac{\boldsymbol{Count}_{\text{HC}}(t)}{\#\text{shder}(t)}} \ , \tag{4.3}$$

where $\#\text{shder}(t)$ is the number of shortest derivations for tree t and

$$P^i_{\text{sh}_\beta}(t) = \beta(f_1(d^i(t))) \cdots \beta(f_{\text{lgth}(d^i(t))}(d^i(t)))$$

is the probability of the i-th shortest derivation $d^i(t)$ for t, consisting of the fragments $f_1(d^i(t)), \ldots, f_{\text{lgth}(d^i(t))}(d^i(t)) \in \mathbf{Frag}_{\text{EC}}$. The side conditions remain the same. Now the term to be maximized is only a product of weights. Rearranging the formula and multiplying together powers of weights of the same fragments ($[\beta(f)]^{e_1} \cdots [\beta(f)]^{e_m} = [\beta(f)]^{e_1 + \cdots + e_m}$), we arrive at the term

$$\arg\max_{\beta \in W} [\beta(f_1)]^{r_1} \cdots [\beta(f_N)]^{r_N} \ , \tag{4.4}$$

where $f_1, \ldots, f_N$ are the fragments involved in the shortest derivations of the parses in HC, and for $k = 1, \ldots, N$:

$$r_k := \sum_{t \in \text{HC}} \frac{\boldsymbol{Count}_{\text{HC}}(t)}{\#\text{shder}(t)} \sum_{i=1}^{\#\text{shder}(t)} \boldsymbol{Count}_{d^i(t)}(f_k) \quad (k = 1, \ldots, N) \,.$$

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 conditions (4.2), we see that each fragment $f \in \mathbf{Frag}_{\text{EC}} \setminus \{f_1, \ldots, f_N\}$ with $\text{root}(f) \in \{R_1, \ldots, R_M\}$ must be assigned the weight $\pi(f) = 0$ in order to maximize the corresponding product in (4.4). Further, we realize that the weights assigned to fragments $f \in \mathbf{Frag}_{\text{EC}}$ with $\text{root}(f) \notin \{R_1, \ldots, R_M\}$ have no influence on the outcome of the maximization problem. We will reserve this weight mass for Part 2 of the estimation procedure by choosing for each $R \notin \{R_1, \ldots, R_M\}$ $\beta(\heartsuit_R) = 1$ and thus $\beta(f) = 0$ for all fragments f with $\text{root}(f) = R$. Since the side conditions for weights of fragments with different roots are independent of each other, we obtain an equivalent maximization problem by splitting the product in (4.4) into a separate optimization problem for every root label $R \in \{R_1, \ldots, R_M\}$ as follows:

$$\underset{\langle \beta(f_j) \rangle_{\text{root}(f_j)=R}}{\arg\max} \prod_{\substack{j \in \{1,\ldots,N\}: \\ \text{root}(f_j)=R}} [\beta(f_j)]^{r_j} \quad , \tag{4.5}$$

where

$$\sum_{j \in \{1,\ldots,N\}:\, \text{root}(f_j)=R} \beta(f_j) = 1 - \beta(\heartsuit_R) \tag{4.6}$$

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

$$\underset{x_1,\ldots,x_n \in \mathbb{R}}{\arg\max}\, x_1^{c_1} \cdots x_n^{c_n} \;, \text{ where } x_1 + \cdots + x_n = c \,,$$

occurring for instance in maximum-likelihood estimation for an unrestricted probability model (cf. Section 2.3) as the special case where $c = 1$. It has the unique solution [2]

$$x_i = c \frac{c_i}{\sum_{k=1}^{n} c_k} \; (i = 1, \ldots, n) \,.$$

Applied to our problem, we thus obtain the solutions

$$\forall\, j \in \{1, \ldots, N\}. \quad \beta(f_j) = \left(1 - \beta\left(\heartsuit_{\text{root}(f_j)}\right)\right) \frac{r_j}{\sum\limits_{\substack{k \in \{1,\ldots,N\}: \\ \text{root}(f_k)=\text{root}(f_j)}} r_k} \,. \tag{4.7}$$

[2] A proof of this can for example be found in [Ney *et al.*, 1997], Subsection 2.4. The proof is given for the case $c = 1$, but goes through in the same way for arbitrary values of c.

4.1.2 The Smoothing Component

Let us come back to the relative frequency mass p_{unkn} of trees in HC that cannot be derived from the fragments in EC. Our aim is to determine from it values for the parameters $\beta(\heartsuit_R)$ and then to distribute this weight mass over the fragments from TC in an appropriate way.

One can regard p_{unkn} as a measure of how well-chosen the currently used set of fragments is. If p_{unkn} is zero, all full parse trees from the held-out corpus were derivable, which means our set of fragments is just perfect. A value greater than zero for p_{unkn} is an indication that we need to enlarge our fragment set. This can be done by involving new fragments from HC and by allocating weight mass to fragments from EC that did not participate in a shortest derivation of an HC-tree and thus are assigned weight zero under the above β-estimation procedure. However, it should be taken care that none of those fragments is assigned more weight than a preferred one, even if p_{unkn} is very high.

As explicated in the previous subsection, $\beta(\heartsuit_R)$ should always be one for roots $R \notin \{R_1, \ldots, R_M\}$ (see Equation (4.5) for a definition of $\{R_1, \ldots, R_M\}$). The same applies to new nonterminals from HC that did not occur in EC at all. The only further requirement to be made such that the consistency proof of DOP* (given in Section 4.2) goes through is that $\beta(\heartsuit_{\text{S}}) \leq p_{\text{unkn}}$. In this thesis, we choose the simple method of assigning for each nonterminal R occurring as a root in the fragment set of the *whole* training corpus TC the weight

$$\beta(\heartsuit_R) := \begin{cases} p_{\text{unkn}} & \text{if } R \in \{R_1, \ldots, R_M\} \text{ (see Eq. (4.5) for a definition of } \{R_1, \ldots, R_M\}) \\ 1 & \text{otherwise} \end{cases} . \tag{4.8}$$

The method of distribution of this weight mass does not affect the consistency of DOP* either. For now, we distribute $\beta(\heartsuit_R)$ in DOP1 fashion over all fragments with root R proportionally to their relative frequencies among the root-R fragments in TC:

$$\forall\, f^* \in \mathbf{Frag}_{\text{TC}}. \quad \beta_{\text{smooth}}(f^*) := \frac{\beta\left(\heartsuit_{\text{root}(f^*)}\right) \; \boldsymbol{Count}_{\mathbf{Frag}_{\text{TC}}}(f^*)}{\boldsymbol{Count}_{\mathbf{Frag}_{\text{TC}}}(\{f \mid \text{root}(f) = \text{root}(f^*)\})} \tag{4.9}$$

Now we can assign the final weights to all fragments f extracted from the training corpus:

$$\forall\, f \in \mathbf{Frag}_{\text{TC}}. \quad \pi(f) := \beta(f) + \beta_{\text{smooth}}(f), \tag{4.10}$$

where for fragments f not occurring in $\mathbf{Frag}_{\text{TC}}$, we define $\beta(f) := 0$. This choice satisfies the condition $\sum_{f:\text{root}(f)=R} \pi(f) = 1$ for all root non-terminals R, since $\sum_{f:\text{root}(f)=R} \beta(f) + \beta(\heartsuit_R) = 1$.

4.1.3 Contemplation

One can regard the DOP* estimator (in which the reserved weight mass is distributed as specified above) as a melange of two components:

1. the core DOP* component—held-out-oriented estimator that bases its weight assignment to fragments on their participation in shortest derivations of held-out parses and

2. a smoothing component—an arbitrary DOP estimator (*e.g.*, DOP1) that serves as a back-up to ensure maximum coverage of the model.

Hereby, the balance between both components is determined by the estimate p_{unkn} of the probability that a full parse tree occurring during testing will be *unknown* in the sense of not being derivable by the held-out component. If the proportion p_{unkn} of unknown parses in HC is very high, DOP* behaves like the back-up estimator. If p_{unkn} is close to 0, DOP* determines the fragment weights according to maximum-likelihood estimation on the held-out corpus HC w.r.t. the model of all DOP probability distributions over **Parses** that result from weight assignments to fragments from EC. (This is in contrast to the standard DOP maximum-likelihood estimator (cf. Section 2.3), which maximizes the likelihood of the *whole* training corpus, and whose probability model is the set of all DOP probability distributions over **Parses** resulting from weight assignments to fragments from again the *whole* training corpus.)

4.2 DOP* is Consistent

In this section, we will show that DOP* is strongly consistent. Remember that an estimator $\varphi : \mathbf{Parses}^* \rightarrow \mathcal{M}_0$ is called strongly consistent w.r.t. a probability model $\mathcal{M}$ if for each $P \in \mathcal{M}$ and each real number $\varepsilon > 0$, we have

$$\lim_{n\rightarrow\infty} \sup_{t\in\mathbf{Parses}} \sum_{\substack{\text{TC}\in\mathbf{Parses}^n: \\ |\varphi_{\text{TC}}(t)-P(t)|\geq\varepsilon}} P(\text{TC}) = 0 \; .$$

As in the case of DOP_{ML}, it turns out that DOP* is not only strongly consistent w.r.t. the probability model $\mathcal{M}_{\text{DOP}}(\text{TC})$ induced by a given training corpus TC (cf. Section 2.3) but even w.r.t. the unrestricted model $\mathcal{M}_0$ of all probability distributions over **Parses**.

The proof is rather involved. We will first demonstrate the intuition behind it using a simple example distribution.

4.2.1 An Example

Let us look back at the toy example given in Figures 2.1, 2.2, and 2.3, for which DOP1 failed. In that example, the training corpus was sampled according to the probability distribution P with $P(t_1) = P(t_2) = 1/2$ and $P(t) = 0$ for all $t \in \mathbf{Parses} \setminus \{t_1, t_2\}$.

The fact that P assigns only a finite number of parses nonzero probabilities makes it fairly easy to show strong consistency of DOP* w.r.t. $\{P\}$: First note that when the size of the training corpus TC goes to infinity, we have with probability arbitrarily close to one that t_1 and t_2 will be contained both in the corresponding extraction portion EC(TC) and the held-out portion HC(TC). Thus, with probability arbitrarily close to one, the proportion $\boldsymbol{rf}_{\mathrm{HC(TC)}}(\{t \in \mathrm{EC(TC)}\})$ of full parse trees in HC(TC) that are contained in EC(TC) will be one. Formally speaking, for each $q > 0$, there is an $N \in \mathbb{N}$ such that for each $n \in \mathbb{N}$ with $n \geq N$, we have

$$\sum_{\substack{\mathrm{TC} \in \mathbf{Parses}^n: \\ \boldsymbol{rf}_{\mathrm{HC(TC)}}(\{t \in \mathrm{EC(TC)}\})=1}} P(\mathrm{TC}) \geq 1 - q \ .$$

If $\boldsymbol{rf}_{\mathrm{HC(TC)}}(\{t \in \mathrm{EC(TC)}\}) = 1$, then the shortest derivation of a full parse tree t occurring in HC(TC) (in our case either t_1 or t_2) is the length-one derivation $\langle t \rangle$ itself. Thus, t_1 and t_2 are the only fragments participating in shortest derivations of trees in HC(TC) and are assigned the r-parameters (cf. Figure 4.1, Step 5)

$$\begin{aligned} r_1 &:= \boldsymbol{Count}_{\mathrm{HC(TC)}}(t_1) \\ r_2 &:= \boldsymbol{Count}_{\mathrm{HC(TC)}}(t_2) \end{aligned}$$

and the β-weights (cf. Figure 4.1, Step 7)

$$\begin{aligned} \beta(t_1) &:= (1 - \beta(\heartsuit_{\mathrm{S}}))\frac{r_1}{r_1 + r_2} = (1 - \beta(\heartsuit_{\mathrm{S}}))\boldsymbol{rf}_{\mathrm{HC(TC)}}(t_1) \ , \\ \beta(t_2) &:= (1 - \beta(\heartsuit_{\mathrm{S}}))\frac{r_2}{r_1 + r_2} = (1 - \beta(\heartsuit_{\mathrm{S}}))\boldsymbol{rf}_{\mathrm{HC(TC)}}(t_2) \ . \end{aligned}$$

All other β-weights are set to zero.

Further, $\boldsymbol{rf}_{\mathrm{HC(TC)}}(\{t \in \mathrm{EC(TC)}\}) = 1$ implies that the proportion p_{unkn} of full parse trees in HC(TC) that are not derivable from $\mathbf{Frag}_{\mathrm{EC(TC)}}$ is zero. By the condition made on the choice of $\beta(\heartsuit_{\mathrm{S}})$, stating that $\beta(\heartsuit_{\mathrm{S}}) \leq p_{\mathrm{unkn}}$ (cf. Subsection 4.1.2), we obtain $\beta(\heartsuit_{\mathrm{S}}) = 0$ and hence for all $t \in \mathbf{Parses}$,

$$\beta(t) = \boldsymbol{rf}_{\mathrm{HC(TC)}}(t) \ .$$

Therefore, we have for $t \in \{t_1, t_2\}$, (cf. Figure 4.1, Step 8),

$$\pi(t) = \beta(t) + \underbrace{\beta_{\mathrm{smooth}}(t)}_{\substack{=0 \text{ since} \\ \beta(\heartsuit_{\mathrm{S}})=0}} = \boldsymbol{rf}_{\mathrm{HC(TC)}}(t)$$

and for all other fragments f with root 'S',

$$\pi(f) = \underbrace{\beta(f)}_{=0} + \underbrace{\beta_{\text{smooth}}(f)}_{\substack{=0 \text{ since} \\ \beta(\heartsuit_{\text{S}})=0}} = 0$$

Thus, the resulting DOP probability distribution is

$$\begin{aligned} \text{DOP}^*_{\text{TC}}(t_1) &= \boldsymbol{rf}_{\text{HC(TC)}}(t_1)\,, \\ \text{DOP}^*_{\text{TC}}(t_2) &= \boldsymbol{rf}_{\text{HC(TC)}}(t_2)\,, \text{ and} \\ \text{DOP}^*_{\text{TC}}(t) &= 0 = \boldsymbol{rf}_{\text{HC(TC)}}(t) \text{ for each } t \in \mathbf{Parses} \setminus \{t_1, t_2\}\,. \end{aligned}$$

We have now established that when the size of the training corpus TC goes to infinity, we have with probability arbitrarily close to one that $\text{DOP}^*_{\text{TC}}(t) = \boldsymbol{rf}_{\text{HC(TC)}}(t)$ for all $t \in \mathbf{Parses}$, *i.e.*, for each $q > 0$, there is an $N \in \mathbb{N}$ such that for each $n \in \mathbb{N}$ with $n \geq N$, we have

$$\sum_{\substack{\text{TC} \in \mathbf{Parses}^n: \\ [\forall\, t \in \mathbf{Parses}\,. \\ \boldsymbol{rf}_{\text{HC(TC)}}(t) = \text{DOP}^*_{\text{TC}}(t)]}} P(\text{TC}) \geq 1 - q\,.$$

From that it seems plausible that we can infer that for each $\varepsilon > 0$ and $q > 0$, there is an $N \in \mathbb{N}$ such that for each $n \in \mathbb{N}$ with $n \geq N$, we have

$$\forall\, t \in \mathbf{Parses}. \quad \sum_{\substack{\text{TC} \in \mathbf{Parses}^n: \\ |\text{DOP}^*_{\text{TC}}(t) - P(t)| \geq \varepsilon}} P(\text{TC}) \leq q$$

since for each t, its relative frequency in HC(TC) approaches $P(t)$ with probability arbitrarily close to one as the size of TC and thus the size of HC(TC) approaches infinity (we will see in the proof below how to make this argument formal). Strong consistency then follows immediately.

4.2.2 The Proof

In the case of a probability distribution P that assigns nonzero probabilities to an infinite number of full parse trees, the reasoning becomes a lot subtler. As a matter of fact, $\boldsymbol{rf}_{\text{HC(TC)}}(\{t \in \text{EC(TC)}\})$ will **not** necessarily become one and p_{unkn} will not necessarily become zero with probability arbitrarily close to one! We will have to argue that with probability arbitrarily close to one, $1 - \boldsymbol{rf}_{\text{HC(TC)}}(\{t \in \text{EC(TC)}\})$ and p_{unkn} become so small that the resulting effect on the expected loss is arbitrarily small.

Theorem 4.2.1 DOP* *is strongly consistent w.r.t. the model* $\mathcal{M}_0$ *of all probability distributions over* **Parses**.

Proof: Let P be a probability distribution over **Parses**. Further, let $\varepsilon > 0$ and $q > 0$ be two real numbers. In order to show strong consistency, we will specify an $N \in \mathbb{N}$ such that for each $n \in \mathbb{N}$ with $n \geq N$, we have

$$\forall\, t \in \mathbf{Parses}. \quad \sum_{\substack{\text{TC} \in \mathbf{Parses}^n: \\ \left|\text{DOP}^*_{\text{TC}}(t) - P(t)\right| \geq \varepsilon}} P(\text{TC}) \leq q \tag{4.11}$$

and thus

$$\sup_{t \in \mathbf{Parses}} \sum_{\substack{\text{TC} \in \mathbf{Parses}^n: \\ \left|\text{DOP}^*_{\text{TC}}(t) - P(t)\right| \geq \varepsilon}} P(\text{TC}) \leq q\ .$$

To establish (4.11), we choose a finite set $T \in \mathbf{Parses}$ such that $\sum_{t' \in T} P(t') \geq 1 - \varepsilon/2$ and $P(t') > 0$ for all $t' \in T$. The choice of such a set is possible since $\sum_{t' \in \mathbf{Parses}} P(t') = 1$. Now define $\varepsilon' := \frac{\varepsilon}{2|T|}$.

In the following, EC(TC) and HC(TC) will denote the actual-training part and the held-out part of the training corpus TC, respectively, according to the fixed method of splitting. Further, n^{EC} and n^{HC} will denote the sizes of the actual-training and the held-out part, respectively, when splitting a training corpus of size n.

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

$$\sum_{\substack{\text{TC} \in \mathbf{Parses}^n: \\ \sum_{\{t' \in T\}} \left|rf_{\text{HC(TC)}}(t') - P(t')\right| \geq \varepsilon'}} P(\text{TC}) \qquad \leq q/2\ .$$

Proof of Claim: It is

$$\begin{aligned}
\sum_{\substack{\text{TC}\in\mathbf{Parses}^n:\\ \underbrace{\sum_{\{t'\in T\}}\left|\boldsymbol{rf}_{\text{HC(TC)}}(t')-P(t')\right|}_{\leq |T|\max_{\{t'\in T\}}\left|\boldsymbol{rf}_{\text{HC(TC)}}(t')-P(t')\right|}\geq\varepsilon'}} P(\text{TC}) &\leq \sum_{\substack{\text{TC}\in\mathbf{Parses}^n:\\ \max_{\{t'\in T\}}\left|\boldsymbol{rf}_{\text{HC(TC)}}(t')-P(t')\right|\geq\frac{\varepsilon'}{|T|}}} P(\text{TC}) \\
&= \sum_{\substack{\text{TC}\in\mathbf{Parses}^n:\\ \exists\, t'\in T \text{ s.t. } \left|\boldsymbol{rf}_{\text{HC(TC)}}(t')-P(t')\right|\geq\frac{\varepsilon'}{|T|}}} P(\text{TC}) \\
&\leq \sum_{t'\in T}\sum_{\substack{\text{TC}\in\mathbf{Parses}^n:\\ \left|\boldsymbol{rf}_{\text{HC(TC)}}(t')-P(t')\right|\geq\frac{\varepsilon'}{|T|}}} P(\text{TC}) \\
&= \sum_{t'\in T}\underbrace{\sum_{\substack{\text{HC}\in(\mathbf{Parses})^{n^{\text{HC}}}:\\ |\boldsymbol{rf}_{\text{HC}}(t')-P(t')|\geq\frac{\varepsilon'}{|T|}}} P(\text{HC})}_{\leq\frac{1}{4n^{\text{HC}}(\varepsilon'/|T|)^2}\text{ by Lemma 3.3.1}} \\
&\leq \frac{|T|^3}{4n^{\text{HC}}(\varepsilon')^2}\,.
\end{aligned}$$

Choosing N_1 large enough ensures

$$\frac{|T|^3}{4n^{\text{HC}}(\varepsilon')^2}\leq\frac{q}{2}$$

for all $n\geq N_1$, since $\lim_{n\to\infty} n^{\text{HC}}=\infty$ by the condition made on the corpus division operation. ◀

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

$$\sum_{\substack{\text{TC}\in\mathbf{Parses}^n:\\ \exists\, t\in T \text{ s.t. } t \text{ occurs in}\\ \text{HC(TC) but not in EC(TC)}}} P(\text{TC})\leq\frac{q}{2}\,.$$

PROOF OF CLAIM: We have

$$\begin{aligned}
\sum_{\substack{\text{TC}\in\mathbf{Parses}^n:\\ \exists\, t\in T \text{ s.t. } t\in\text{HC(TC)}\\ \text{and } t\notin\text{EC(TC)}}} P(\text{TC}) \;&\leq\; \sum_{\substack{\text{TC}\in\mathbf{Parses}^n:\\ \exists\, t\in T \text{ s.t. } t\notin\text{EC(TC)}}} P(\text{TC}) \\
&\leq\; \sum_{t'\in T} \underbrace{\sum_{\text{TC}\in\mathbf{Parses}^n:t'\notin\text{EC(TC)}} P(\text{TC})}_{\substack{=P(\text{“}t' \text{ was drawn in none of the } n^{\text{EC}}\\ \text{samples of the extraction corpus”})\\ =\left[1-P(t')\right]^{n^{\text{EC}}}}} \\
&=\; \sum_{t'\in T}\left[1-P(t')\right]^{n^{\text{EC}}} \leq |T|\underbrace{[1-\min_{t'\in T}\underbrace{P(t')}_{>0}]^{n^{\text{EC}}}}_{<1} \\
&\leq\; q/2 \text{ for all } n\geq N_2 \text{ if } N_2 \text{ chosen appropriately.}
\end{aligned}$$

◀

CLAIM 3 Let TC be a training corpus and t a full parse tree. Assume that we have $\sum_{\{t'\in T\}}\left|\boldsymbol{rf}_{\text{HC(TC)}}(t')-P(t')\right|<\varepsilon'$ and $|\text{DOP}^*_{\text{TC}}(t)-P(t)|\geq\varepsilon$. Then there is a $t''\in T$ s.t. t'' occurs in HC(TC) but not in EC(TC).

PROOF OF CLAIM: Assume indirectly that we have

$$\sum_{\{t'\in T\}}\left|\boldsymbol{rf}_{\text{HC(TC)}}(t')-P(t')\right|<\varepsilon' \tag{4.12}$$

and

$$|\text{DOP}^*_{\text{TC}}(t)-P(t)|\geq\varepsilon \tag{4.13}$$

but that all trees in T that occur in HC(TC) also occur in EC(TC). Then these trees, in the following denoted by $t_1,\ldots,t_m$, occur also as fragments in $\mathbf{Frag}_{\text{EC(TC)}}$. Thus, for each such tree, its shortest derivation from $\mathbf{Frag}_{\text{EC(TC)}}$ is the unique length-1-derivation consisting only of the tree itself.

Since each derivation of a full parse tree in HC(TC) contains exactly one fragment with root 'S', namely the first fragment of the derivation (remember that 'S' is only allowed to occur as root node in a full parse tree and thus can only occur as root node in a fragment), it is easy to see that the r-parameters (cf. Figure 4.1, Step 5) that DOP* assigns to the root-'S' fragments involved in shortest derivations sum up to the number of full parse trees in the held-out corpus that are derivable from the fragments in $\mathbf{Frag}_{\text{EC(TC)}}$, which is equal to $(1-p_{\text{unkn}})\,|\text{HC(TC)}|$. Thus each t_j $(j=1,\ldots,m)$ is assigned the β-weight (cf. Figure 4.1, Step 7)

$$\beta(t_j)=(1-\beta(\heartsuit_{\text{S}}))\,\frac{\boldsymbol{Count}_{\text{HC(TC)}}(t_j)}{(1-p_{\text{unkn}})\,|\text{HC(TC)}|}\;.$$

By the condition made on the choice of $\beta(\heartsuit_{\mathrm{S}})$, stating that $\beta(\heartsuit_{\mathrm{S}}) \leq p_{\mathrm{unkn}}$ (cf. Subsection 4.1.2), we obtain

$$\beta(t_j) \geq \frac{\boldsymbol{Count}_{\mathrm{HC(TC)}}(t_j)}{|\mathrm{HC(TC)}|} = \boldsymbol{rf}_{\mathrm{HC(TC)}}(t_j) \;\; (j = 1, \ldots, m)$$

and thus (cf. Figure 4.1, Step 8)

$$\mathrm{DOP}^*_{\mathrm{TC}}(t_j) \geq \pi(t_j) \geq \beta(t) \geq \boldsymbol{rf}_{\mathrm{HC(TC)}}(t_j) \;\; (j = 1, \ldots, m).$$

Since full parse trees $t' \in T$ not occurring in HC(TC) (*i.e.*, for which $\boldsymbol{rf}_{\mathrm{HC(TC)}}(t') = 0$ holds) trivially satisfy $\mathrm{DOP}^*_{\mathrm{TC}}(t') \geq \boldsymbol{rf}_{\mathrm{HC(TC)}}(t')$, we have

$$\forall\, t' \in T.\ \mathrm{DOP}^*_{\mathrm{TC}}(t') \geq \boldsymbol{rf}_{\mathrm{HC(TC)}}(t') \;\; .$$

Since (4.12) implies $\left|\boldsymbol{rf}_{\mathrm{HC(TC)}}(t') - P(t')\right| < \varepsilon'$ and thus $\boldsymbol{rf}_{\mathrm{HC(TC)}}(t') > P(t') - \varepsilon'$ for all $t' \in T$, it follows

$$\forall\, t' \in T.\ \mathrm{DOP}^*_{\mathrm{TC}}(t') > P(t') - \varepsilon' \;\; . \tag{4.14}$$

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

$$\begin{aligned}
\sum_{t' \in T \setminus \{t''\}} \mathrm{DOP}^*_{\mathrm{TC}}(t') \;\; &> \;\; \sum_{t' \in T \setminus \{t''\}} (P(t') - \varepsilon') \\
&= \underbrace{\sum_{t' \in T} P(t')}_{\substack{\geq 1-\varepsilon/2 \\ \text{by Def. of } T}} - P(t'') - \underbrace{(|T| - 1)\varepsilon'}_{\substack{\leq \varepsilon/2 \\ \text{by Def. of } \varepsilon'}} \geq 1 - \varepsilon - P(t'') \;\; .
\end{aligned}$$

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

$$\begin{aligned}
\mathrm{DOP}^*_{\mathrm{TC}}(t'') \;\; &= \;\; 1 - \sum_{t' \in \mathbf{Parses} \setminus \{t''\}} \mathrm{DOP}^*_{\mathrm{TC}}(t') \\
&\leq \;\; 1 - \sum_{t' \in T \setminus \{t''\}} \mathrm{DOP}^*_{\mathrm{TC}}(t') \\
&< \;\; 1 - (1 - \varepsilon - P(t'')) = P(t'') + \varepsilon \;\; .
\end{aligned}$$

Together with (4.14) this yields

$$\forall\, t'' \in T.\ |\mathrm{DOP}^*_{\mathrm{TC}}(t'') - P(t'')| < \varepsilon \;\; . \tag{4.15}$$

If we can show the same result for full parse trees $t'' \notin T$, we are done, since that means that (4.13) cannot be fulfilled for any full parse tree t, which is a

contradiction. For that purpose, we derive from (4.14), this time by summing up over all $t' \in T$,

$$\begin{aligned}\sum_{t' \in T} \mathrm{DOP}^*_{\mathrm{TC}}(t') &> \sum_{t' \in T} (P(t') - \varepsilon') \\ &= \underbrace{\sum_{t' \in T} P(t')}_{\substack{\geq 1-\varepsilon/2 \\ \text{by Def. of } T}} - \underbrace{|T|\varepsilon'}_{\substack{=\varepsilon/2 \\ \text{by Def. of } \varepsilon'}} \geq 1 - \varepsilon \ .\end{aligned}$$

Thus we have

$$\begin{aligned}\forall\, t'' \in (\mathbf{Parses} \setminus T).\ \mathrm{DOP}^*_{\mathrm{TC}}(t'') &\leq 1 - \textstyle\sum_{t' \in T} \mathrm{DOP}^*_{\mathrm{TC}}(t') \\ &< 1 - (1 - \varepsilon) = \varepsilon \leq P(t'') + \varepsilon \ .\end{aligned} \tag{4.16}$$

Further, it holds that

$$\begin{aligned}\forall\, t'' \in (\mathbf{Parses} \setminus T).\ P(t'') - \varepsilon &\leq 1 - \underbrace{\sum_{t' \in T} P(t')}_{\substack{\geq 1-\varepsilon/2 \\ \text{by Def. of } T}} - \varepsilon \\ &\leq -\varepsilon/2 \\ &< \mathrm{DOP}^*_{\mathrm{TC}}(t'') \ ,\end{aligned}$$

which together with (4.16) yields

$$\forall\, t'' \in (\mathbf{Parses} \setminus T).\ |\mathrm{DOP}^*_{\mathrm{TC}}(t'') - P(t'')| < \varepsilon \ ,$$

and thus with (4.15),

$$\forall\, t'' \in \mathbf{Parses}.\ |\mathrm{DOP}^*_{\mathrm{TC}}(t'') - P(t'')| < \varepsilon \ .$$

As indicated above, this leads to the desired contradiction, since (4.13) cannot be fulfilled for any full parse tree t. ◀

Now we are finally able to specify the required $N \in \mathbb{N}$ such that for all natural numbers $n \geq N$, (4.11) 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 $t \in \mathbf{Parses}$ and $n \in \mathbb{N}$ with $n > N$,

$$\begin{aligned}
\sum_{\substack{\text{TC}\in\mathbf{Parses}^n:\\ \left|\text{DOP}^*_{\text{TC}}(t)-P(t)\right|\geq\varepsilon}} P(\text{TC}) \;=\; & \underbrace{\sum_{\substack{\text{TC}\in\mathbf{Parses}^n:\\ \sum_{\{t'\in T\}}\left|rf_{\text{HC}(\text{TC})}(t')-P(t')\right|\geq\varepsilon'\\ \text{and } \left|\text{DOP}^*_{\text{TC}}(t)-P(t)\right|\geq\varepsilon}} P(\text{TC})}_{\leq q/2 \text{ by Claim 1}} \;+\; \underbrace{\sum_{\substack{\text{TC}\in\mathbf{Parses}^n:\\ \sum_{\{t'\in T\}}\left|rf_{\text{HC}(\text{TC})}(t')-P(t')\right|<\varepsilon'\\ \text{and } \left|\text{DOP}^*_{\text{TC}}(t)-P(t)\right|\geq\varepsilon}} P(\text{TC})}_{\leq \sum_{\substack{\text{TC}\in\mathbf{Parses}^n:\\ \exists\, t''\in T \text{ s.t. } t'' \text{ occurs in}\\ \text{HC(TC) but not in EC(TC)}}} P(\text{TC}) \text{ by Claim 3}} \\
\leq\; & q/2 \;+\; \underbrace{\sum_{\substack{\text{TC}\in\mathbf{Parses}^n:\\ \exists\, t''\in T \text{ s.t. } t'' \text{ occurs in}\\ \text{HC(TC) but not in EC(TC)}}} P(\text{TC})}_{\leq q/2 \text{ by Claim 2}} \\
\leq\; & q\,.
\end{aligned}$$

QED

4.3 The Number of Extracted Fragments

Note that the version of DOP* explained so far assigns nonzero weights to *all* fragments from the training corpus, since the second part of the estimation procedure distributes the reserved weight mass equally over all fragments.

In the following, we will show that the core estimator that assigns the β-weights actually extracts a number of fragments that is linear in the number of occurrences of depth-one fragments of $\mathbf{Frag}_{\text{HC}}$, and thus, the number of nodes in HC. This is in strong contrast to DOP1, which extracts all possible fragments, *i.e.*, exponentially many in the number of nodes in HC. The proof relies on the assumption that the average number of shortest derivations of a held-out parse is limited by a constant N. This assumption turns out to be unproblematic in practice; the average number of shortest derivations of held-out parses was less than 3.5 in all our experiments.

For each held-out parse, the β-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 of the depth-one fragments contained in t. Therefore, the number of fragments extracted from $\mathbf{Frag}_{\text{EC}}$ for each shortest derivation of a parse $t \in$ HC is bounded by the number of depth-one fragment occurrences (and hence, the number of nodes) in t. Thus overall the procedure extracts at most N times the number of depth-one fragment occurrences (and hence, the number of nodes) in the held-out corpus.

In practice, we will make use of this finding by not distributing the reserved weight mass over all fragments, as explained in Subsection 4.1.2, but rather only

over the depth-one fragments, in order to obtain an estimator that is much more efficient than DOP1. Another possibility, which we also tested, is to not reserve any weight mass at all (*i.e.*, setting $\beta(\heartsuit_R) := 0$ for all nonterminals R) and to back off to a different parser whenever no parse can be found for a test sentence.

4.4 DOP* Is *Not* Biased Towards Fragments of Large Parse Trees

As we have seen in Subsection 2.2.4, the original estimator DOP1 is biased towards fragments of large parse trees. Since DOP* is consistent and thus unbiased in the limit, DOP* does not have this problem when the size of the training data approaches infinity (in contrast to DOP1, which keeps that bias with growing training data). But how does DOP* behave when the training data is small?

Let us assume that p_{unkn} turns out to be small, so that the core (held-out estimation) component of DOP* clearly dominates. (Our experiments—cf. Chapter 5—will justify this assumption.) Then only the fragments in $\mathbf{Frag}_{\text{EC}}$ are assigned significant weight mass. Now let t_1 be a large and t_2 a small full parse tree occurrence in EC, such that we have considerably more root-'S' fragments from t_1 than from t_2 in $\mathbf{Frag}_{\text{EC}}$. To keep matters simple, assume that t_1 and t_2 have no fragments in common and that other full parse tree occurrences in EC do not contain any fragments from t_1 or t_2 either. Then DOP1 would assign all fragments from t_1 and t_2 with the same root equal weights and thus clearly favor full parse trees derived from t_1-fragments.

DOP*, on the other hand, bases its weight assignment to fragments on their participation in the derivations of trees in HC. Assume that parse trees whose shortest derivations contain a t_1-fragment occur about equally often in HC as parse trees whose shortest derivations contain a t_2-fragment do. Since only those fragments actually participating in shortest derivations are assigned nonzero β-weights (and since the weights assigned to them are proportional to their number of occurrence in shortest derivations[3]), the sum of the β-weights assigned to the t_1-fragments with a certain root will roughly equal the sum of the β-weights assigned to the t_2-fragments with the same root . Since p_{unkn} is small, the same will hold for the final π-weights. If the assumption made at the beginning of this paragraph does not apply, that will be an indication that either one of the trees t_1 and t_2 proved to be of more importance during testing on the held-out data, and thus the fragments of that tree are rightly favored.

[3]Here we are actually also assuming that the roots of those fragments are unique in the respective derivations, which is always the case for fragments with root 'S' and often for other fragments.

4.5 Summary

In this chapter, we devised a DOP estimator that fulfills the property of consistency. Furthermore, the estimator extracts only a number of fragments that is linear in the total number of nodes of the trees in the training corpus, thereby circumventing the inefficiency problems of original DOP without giving up on the idea of using arbitrary-size fragments. We also demonstrated that in contrast to DOP1, our estimator does not suffer from the fragment bias problem.

Chapter 5

Empirical Results

In this chapter, we substantiate the theoretical findings of this thesis with empirical evidence.

The experiments were carried out on the Dutch language OVIS corpus [Scha *et al.*, 1996], 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 [Sima'an, 1999, Sima'an & Buratto, 2003]. Parsing OVIS is relatively easy (compared to, *e.g.*, parsing the ATIS or the Wall Street Journal Corpus). About thirty percent of the corpus are one-word utterances; those are usually removed from the testing data in order to decrease variance in results between different training/testing splits. One-word sentences excluded, the average length of an OVIS sentence is 4.6.

5.1 Practical Issues

Since the OVIS corpus is rather small, the issue of how to divide the training corpus into extraction and held-out portion becomes crucial. On the one hand, only fragments from the extraction corpus EC are considered for shortest derivations in the held-out corpus HC. One would wish one could use *all* fragments from TC as EC-fragments. On the other hand, each parse tree in HC yields (in form of its shortest derivations from EC) valuable information on which fragments of EC are relevant for natural language. One would wish one could use *all* fragments from TC as HC-fragments. We do exactly this. Following a suggestion by Sima'an (personal communication), we apply a method similar to *deleted estimation* [Jelinek, 1985]. For this purpose, we split the training corpus into ten equal

portions and run the DOP* estimation algorithm ten times, using successively one of the portions as held-out corpus and the other nine as extraction corpus. Subsequently, we interpolate the ten resulting DOP* weight assignments.

This way, each parse of the training corpus will be used as a held-out parse some time during training with its shortest derivation(s) derived from the other 90 percent of the training corpus. Thus more fragments have the chance to be involved in shortest derivations, which enhances the coverage of the model.

Note that the resulting estimator is still consistent: From Section 4.2 we know for each of the ten provisional estimators that the weight assigned to a full parse tree $t \in$ TC (regarded as a fragment) approaches t's relative frequency in TC when the size of TC goes to infinity. Thus the same must happen for the convex combination of those weights, the weight that the final estimator assigns to t.

Also, it is easy to see that the number of fragments extracted is still linear in the number of nodes in the training corpus: Since each parse in the training corpus will be used exactly once as a held-out parse, the total number of fragments extracted is now bounded by N times the number of nodes in the *whole* training corpus, where N is again the upper bound on the average number of shortest derivations of a held-out parse.

5.2 Testing

Unless noted otherwise, 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. All one-word utterances were ignored in evaluation.

For both DOP1 and DOP* experiments, we used Khalil Sima'an's DOPDIS parser, which is publicly available at `staff.science.uva.nl/~simaan/dopdis`.

5.2.1 The DOP* Variants Used

In the experiments, we used two different variants of DOP*. The first implementation is a pure held-out estimator without smoothing component, which is obtained by setting $\beta(\heartsuit_R) := 0$ for all nonterminals R (cf. Section 4.1). Instead of smoothing, our parser just backs off to DOP1 whenever no parse can be found for a test sentence.[1] During testing, around 9% of the sentences turned out to be unknown to the core DOP* estimator. However, we do not actually need to back off in all those cases. Instead, before backing off to DOP1, we use a PCFG parser trained on the same data to see whether the test sentence is

[1]Note that, in principle, also a non-DOP parser could be used as back-off, which could improve performance by profiting from complementary parsing approaches.

at all parsable.[2] If no parse for the sentence can be generated from the PCFG grammar, there is no avail in consulting DOP1 as the tree languages generated by PCFG and DOP1 grammars acquired from the same treebank are identical. Using this method, back-off to DOP1 had to be performed about 3.6% of the cases. Out of those back-offs, about two thirds were successful in the sense that DOP1's most probable parse was the correct one. In the following, we will call this DOP* implementation $\text{DOP}^*_{\text{B/O}}$.

The second DOP* version implemented, in the following called $\text{DOP}^*_{\text{PCFG}}$, smoothes the core held-out estimator with the PCFG (depth-one) fragments from TC and with the fragments up to depth three of unknown held-out parses (*i.e.*, parse trees in the held-out corpus HC that were not derivable from EC-fragments).

5.2.2 Effects of Inconsistent Estimation

We compare DOP* to DOP1 for different maximum-depth constraints on extracted fragments. Figure 5.1 shows the exact match (EM) rate (number of correctly parsed sentences divided by total number of sentences) for DOP1, $\text{DOP}^*_{\text{B/O}}$, and $\text{DOP}^*_{\text{PCFG}}$ w.r.t. maximum fragment depth (where $\text{DOP}^*_{\text{B/O}}$ backs off to the DOP1 estimator of the corresponding maximum-depth).[3] The $\text{DOP}^*_{\text{B/O}}$ and $\text{DOP}^*_{\text{PCFG}}$ results are strikingly similar.

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 graph in Figure 5.1 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 fragments and depth-five fragments are included. DOP*'s EM rate begins to outperform DOP1's EM rate at depth three. At no depth level, however, the difference in performance was statistically significant.

5.2.3 Learning Curves

In order to compare the learning behavior of DOP1 and DOP* and to determine whether the size of the training corpus was sufficient for an optimal parsing per-

[2]For that purpose, we merely need to use the Viterbi Parsing Algorithm, whose runtime for PCFGs is negligible compared to the Montecarlo Algorithm used to compute the most probable parse of a DOP sentence.

[3]Due to the enormous compilation and testing times for DOP1 at depth five, we were not able to obtain results for DOP1 (and therefore, $\text{DOP}^*_{\text{B/O}}$) for this depth level. Thanks to Nguyen Thuy Linh for providing us with her results on the standard splittings.

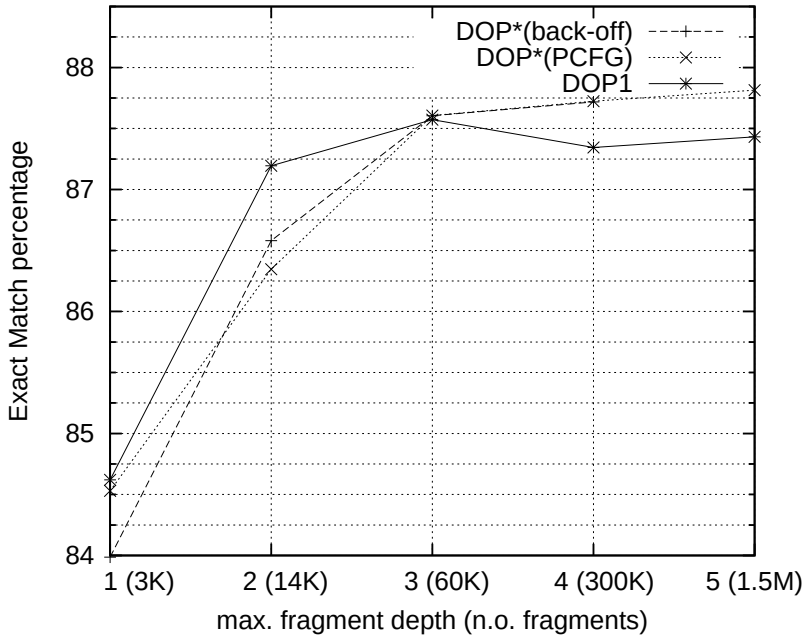


Figure 5.1: Performance for different maximum-depths of extracted fragments.

formance, we randomly removed utterances from the training corpus of our first standard splitting—1000 at a time in eight consecutive steps—and trained DOP1 and $\text{DOP}^*_{\text{PCFG}}$ on the resulting corpora. Parsing performance on the corresponding testing corpus is displayed in Figure 5.2. Both estimator's performance monotonically increases with growing corpus size. Remarkably, DOP1 outperforms $\text{DOP}^*_{\text{PCFG}}$ up to corpus size 8000. Only when provided with the whole training corpus, $\text{DOP}^*_{\text{PCFG}}$ manages to beat DOP1. DOP^* is thus more contingent on the size of the training data, which can be explained by its principle of dividing the training data into extraction and held-out parts: Although the application of deleted estimation ensures that all training samples participate in held-out testing, the size of the extraction corpus is only 9/10 of the whole training corpus at each point of time.

As can also be seen in Figure 5.2, the OVIS corpus is not sufficiently large to ensure optimal parsing performance for DOP: Both estimators' performance still increases when the corpus size is enlarged from 8000 to 9000 utterances. Hereby, $\text{DOP}^*_{\text{PCFG}}$'s increase is stronger, suggesting that $\text{DOP}^*_{\text{PCFG}}$ might have outperformed DOP1 even clearer if more training data had been available.

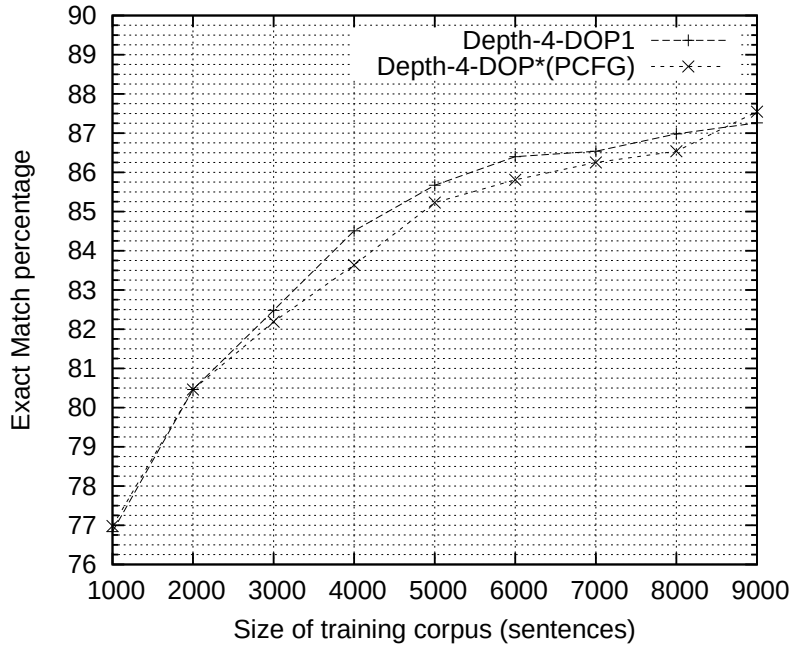


Figure 5.2: Learning behavior on an incrementally increasing training corpus (depth-level four).

5.2.4 Efficiency

Our tests confirmed the anticipated exponential speed-up in testing time, as Table 5.1 shows. These data are in line with Figures 5.4 and 5.3, displaying the number of extracted fragment *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
$\text{DOP}^*_{\text{B/O}}$	5	6	6	18	N/A
$\text{DOP}^*_{\text{PCFG}}$	5	6	6	14	17

Table 5.1: Parsing time for whole testing corpus in minutes.

5.2.5 Other Results

The proportion p_{unkn} of held-out parses (parses of one-word sentences included) that were not derivable from EC-fragments during training (cf. Section 4.1) was always around 8% in our experiments. Figure 5.5 displays the average number

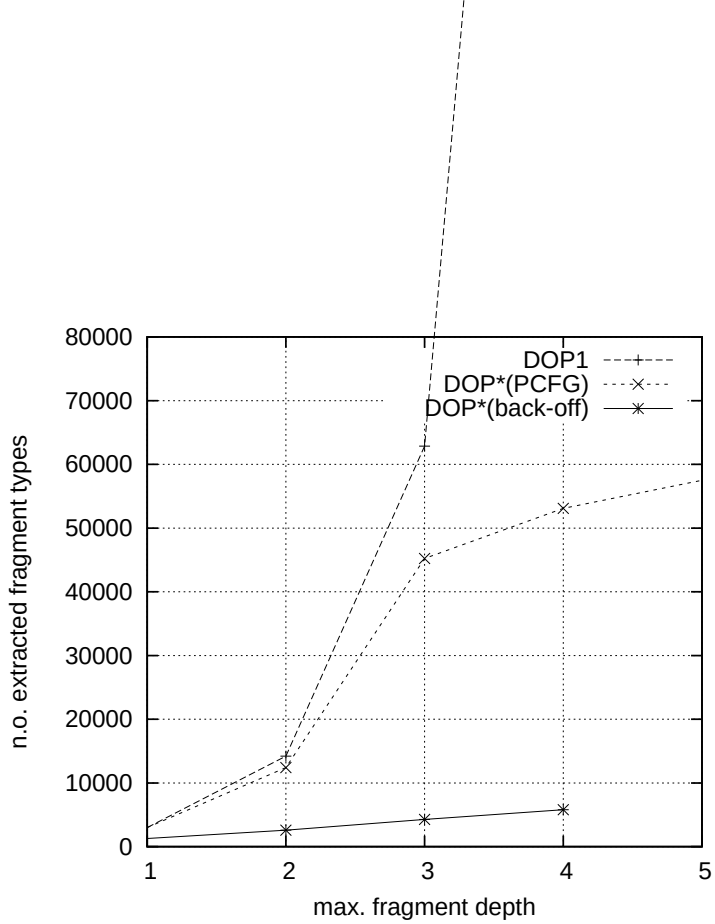


Figure 5.3: DOP*'s number of extracted fragment types for different maximum-depth constraints (linear scale). Unless you printed this thesis on A0 paper, you will only see part of DOP1's vertiginous extraction curve.

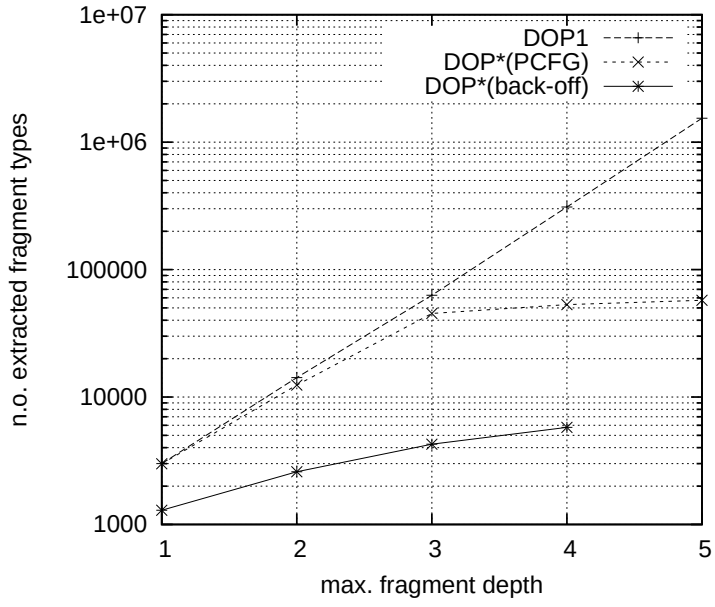


Figure 5.4: The number of extracted fragment types for different maximum-depth constraints (y-logarithmic scale).

of shortest derivations per held-out parse for different maximum-depths. Since about 8% of the held-out parses were not derivable from the extraction corpus and each depth-one parse has a unique derivation, that number averages at 0.92 for depth one. Depth-two experiments resulted in the highest number in the series (3.06).

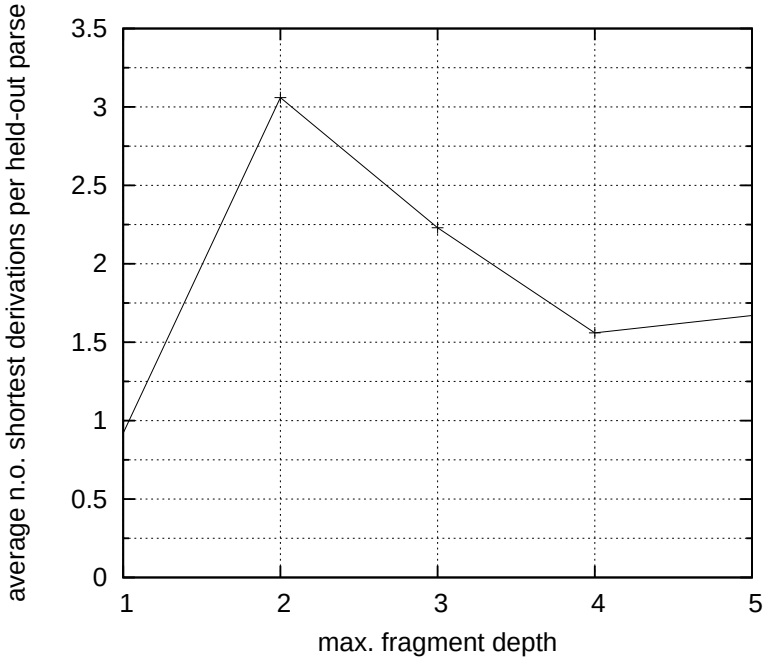


Figure 5.5: Average number of shortest derivations per held-out parse for different depth-levels.

5.3 Summary

In this chapter, we tested two different DOP* implementations against DOP1 on the OVIS corpus. Both implementations outperformed DOP1 in parsing accuracy when fragments of at least depth three were included. Also, their parsing accuracy monotonically increased with increasing maximum fragment depth. In contrast, due to DOP1's negligence of interdependencies between overlapping fragments, DOP1's parsing accuracy actually decreased when fragments of depth greater than three were included.

We also saw that the size of the treebank is crucial for DOP*. Our estimator requires redundancy in the training data to improve over common estimators.

We were able to verify the anticipated exponential speed-up in parsing time empirically. At depth five, the difference between the estimators became drastic:

While DOP* needed only 17 minutes for the whole testing corpus, DOP1 took more than 24 hours.

Chapter 6

Conclusions and Directions for Further Research

In this thesis, we presented a consistent and efficient estimator for the Data-Oriented Parsing model. The estimator has a clear theoretical motivation in a generalization of the maximum-likelihood principle to held-out estimation. Moreover, it achieves an exponential reduction in the number of fragments extracted from the training corpus w.r.t. the common DOP estimator. It might be of interest to investigate whether the principles behind our estimator DOP* could be adapted to DOP-related paradigms such as LFG-DOP or Data-Oriented Translation. Further, we believe that DOP* could be "married" with Back-off DOP as follows: In a first training step, DOP* could be employed to find out which fragments from the training corpus are at all *relevant*. Since DOP* extracts only a fraction of all possible fragments, such a set would be a practically feasible starting point for Back-off DOP. Now back-off re-estimation could proceed, using DOP*'s weight assignments to the fragments as the *given* distributions. However, the current shortcomings of Back-off DOP need first to be addressed.

We empirically validated the theoretical properties of DOP* on the OVIS treebank. Future work should compare DOP* with other existing (non-DOP) parsers. For this purpose, other treebanks (*e.g.*, the Penn Wallstreet Journal Treebank) should be employed.

Tying up to previous work in [Johnson, 2002], we adapted the framework of estimation theory—including two different possible notions of consistency—to statistical parsing and gave a consistency proof for DOP* that can serve as the basis for consistency proofs for other estimators in statistical parsing. Future work might refine the notion of consistency by accounting for the *speed* of the convergence. For each positive real value ε, a consistent estimator would then go along with a minimum sample size $N(\varepsilon)$ needed to ensure an expected loss less than ε w.r.t. all probability distributions in the model. A similar notion exists in computational learning theory.

We also saw that the estimation-theoretic property of *bias* is not interesting for DOP and, in fact, most statistical parsing models. New formal properties of estimators should be defined and studied. For example, the property of *overfitting*, known from machine learning, should be incorporated.

Bibliography

[Baker, 1979] Baker, J. K. 1979. Trainable Grammars for Speech Recognition. *Proc. of Spring Conference of the Acoustical Society of America.*

[Bod, 1991] Bod, Rens. 1991. Data Oriented Parsing. *Proceedings COLING-91*, Amsterdam, The Netherlands.

[Bod & Scha, 1996] Bod, Rens, and Remko Scha. 1996. *Data-Oriented Language Processing: An Overview.* Research report nr. LP-96-13, ILLC Research reports, University of Amsterdam. Available at `www.essex.ac.uk/linguistics/clmt/papers/dop/bodscha.ps`

[Bod, 1998] Bod, Rens. 1998. *Beyond Grammar: An Experience-Based Theory of Language.* Stanford, CA: CSLI Publications.

[Bod, 2000] Bod, Rens. 2000. Combining Semantic and Syntactic Structure for Language Modeling. *Proceedings ICSLP-2000*, Beijing, China. Available at `staff.science.uva.nl/~rens`.

[Bod, 2000] Bod, Rens. 2000. Parsing with the Shortest Derivation. *Proceedings COLING-2000.* Saarbruecken, Germany. Available at `staff.science.uva.nl/~rens`.

[Bod, 2001] Bod, Rens. 2001. What is the Minimal Set of Fragments that Achieves Maximal Parse Accuracy? *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'2001).* Toulouse, France. Available at `staff.science.uva.nl/~rens`.

[Booth, 1969] Booth, T. 1969. Probabilistic representation of for- mal languages. In *Tenth Annual IEEE Symposium on Switching and Automata Theory*, October.

[Bonnema, 2003] Bonnema, Remko. 2003. Probability Models for DOP. In: Bod, R., Scha, R., and Sima'an, K. *Data Oriented Parsing.* CSLI Publications, Stanford University. Stanford, California, USA.

[Bonnema *et al.*, 1999] Bonnema, Remko, Paul Buying, and Remko Scha. 1999. A New Probability Model for Data-Oriented Parsing. *Proceedings of the Amsterdam Colloquium 1999*. Amsterdam. Available at `citeseer.nj.nec.com/bonnema99new.html`.

[Charniak, 1993] Charniak, Eugene. 1993. Statistical Language Learning. Cambridge, MA: MIT Press.

[DeGroot & Schervish, 2002] DeGroot, Morris H., and Mark J. Schervish. 2002. *Probability and Statistics*. 3rd edition. Addison-Wesley.

[De Pauw, 1999] De Pauw, Guy. 1999. Pattern-matching aspects of Data-Oriented Parsing. *Presented at Computational Linguistics in the Netherlands (CLIN)*. Utrecht, Netherlands.

[De Pauw, 2000] De Pauw, Guy. 2000. Aspects of Pattern-Matching in DOP. *Proceedings of the 18th International Conference of Computational Linguistics (COLING 2000)*. Saarbrücken, Germany.

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

[Goodman, 1998] Goodman, Joshua. 1998. *Parsing Inside-Out*. Ph.D. thesis, Harvard University, Massachusetts. Available at `http://citeseer.nj.nec.com/article/goodman98parsing.html`

[Jelinek, 1985] Jelinek, Fred, and Mercer, Robert. 1985. Probability distribution estimation from sparse data. IBM *Technical Disclosure Bulletin* 28:2591-2594.

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

[Johnson, 2002] Johnson, Mark. 2002. The DOP Estimation Method Is Biased and Inconsistent. *Computational Linguistics* 28(1), pages 71-76. Available at `cog.brown.edu/~mj/Publications.htm`.

[Krenn & Samuelsson, 1997] Krenn, Brigitte, and Christer Samuelsson. 1997. *The Linguist's Guide to Statistics—Don't Panic*. `citeseer.nj.nec.com/krenn97linguists.html`.

[Lari & Young, 1990] Lari, K., and S. J. Young. 1990. The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech and Language*, 4:35-56.

[Manning & Schütze, 1999] Manning, Christopher, and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts.

[Ney *et al.*, 1997] Ney, Hermann, Sven Martin, and Frank Wessel. 1997. Statistical Language Modeling Using Leaving-One-Out. In: Steve Young and Gerrit Bloothooft (eds.), *Corpus-based Methods in Language and Speech Processing*, pp. 174-207. Kluwer Academic, Dordrecht.

[Nguyen, 2004] Nguyen, Thuy Linh. 2004. A Fragment-Based Estimator for Data-Oriented Parsing. Master's Thesis. Institute for Logic, Language and Computation, University of Amsterdam.

[Pereira & Schabes, 1992] Pereira, Fernando, and Yves Schabes. 1992. Inside-out reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*. Berkeley, CA.

[Prescher *et al.*, 2004] Prescher, Detlef, Remko Scha, Khalil Sima'an, and Andreas Zollmann. 2004. On the Statistical Consistency of DOP Estimators. To be published elsewhere. Available at `http://staff.science.uva.nl/~azollman/publications.html`.

[Prescher, 2003] Prescher, Detlef. 2003. *A Tutorial on the Expectation-Maximization Algorithm Including Maximum-Likelihood Estimation and EM Training of Probabilistic Context-Free Grammars*. Presented at ESSLLI-03, Vienna, Austria. Available at `http://staff.science.uva.nl/~prescher/papers/`.

[Scha, 1990] Scha, Remko. 1990. Taaltheorie en Taaltechnologie; Competence en Performance. In: de Kort, Q. A. M., and Leerdam, G. L. J., (eds.), *Computertoepassingen in de Neerlandistiek*, Almere: Landelijke Vereniging van Neerlandici (LVVN-jaarboek). English translation as: Language Theory and Language Technology; Competence and Performance; `http://iaaa.nl/rs/LeerdamE.html`

[Scha *et al.*, 1996] Scha, Remko, Remko Bonnema, Rens Bod, and Khalil 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/`.

[Siegrist, 2004] Siegrist, Kyle. 2004. *Virtual Laboratories in Probability and Statistics*. `www.math.uah.edu/stat`.

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

[Sima'an & Buratto, 2003] Sima'an, Khalil, and Luciano Buratto. Backoff Parameter Estimation for the DOP Model. *Proceedings of the European Conference on Machine Learning (ECML'03)*. Dubrovnik, Croatia. Available at `staff.science.uva.nl/~simaan`

Index