Backoff Parameter Estimation for the DOP Model

Khalil Sima'an and Luciano Buratto*

Institute for Logic, Language and Computation (ILLC)
University of Amsterdam, The Netherlands
simaan@science.uva.nl; lburatto@citri.iq.usp.br

Abstract. The Data Oriented Parsing (DOP) model currently achieves state-of-the-art parsing on benchmark corpora. However, existing DOP parameter estimation methods are known to be biased, and ad hoc adjustments are needed in order to reduce the effects of these biases on performance. In contrast with earlier work, in this paper we show that the DOP parameters constitute a hierarchically structured space of correlated events (rather than a set of disjoint events). The correlations between the different parameters can be expressed by an asymmetric relation called 'backoff". Subsequently, we present a novel recursive estimation algorithm that exploits this hierarchical structure for parameter estimation through discounting and backoff. Finally, we report on experiments showing error reductions of up to 15% in comparison to earlier estimation methods.

1 Introduction

The Data Oriented Parsing (DOP) model currently exhibits state-of-the-art performance on benchmark corpora [1]. A DOP model is trained on a treebank[1] by extracting all subtrees of the treebank trees and employing them as the basic rewrite events (or productions) of a formal grammar. The problem of how to estimate the probabilities of the subtrees from the treebank turns out *not* as straightforward as originally thought. So far, there exist three suggestions for parameter estimation [2,3,1]. As shown in [3,4] and in the present paper, all three estimation procedures turn out to be biased in an unintuitive manner. Therefore, the problem of how to estimate the DOP parameters in a productive, yet computationally reasonable manner, remains unsolved.

Parameter estimation for DOP is complex due to two unique aspects of the model: (1) a parse-tree in DOP is often generated through multiple, different rewrite derivations, and (2) the model consists of treebank subtrees of arbitrary size. These two aspects distinguish DOP from other existing models that are predominantly based on the paradigm of History-Based Stochastic Models (HBSG)

* We thank Rens Bod, Detlef Prescher and the reviewers for their comments.

[1] A treebank is a sample of parse-sentence pairs drawn from a domain of language use; the parses are the correct syntactic structures as perceived by humans.

N. Lavrač et al. (Eds.): ECML 2003, LNAI 2837, pp. 373–384, 2003.

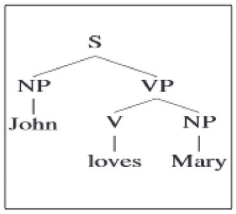

Fig. 1. A toy treebank

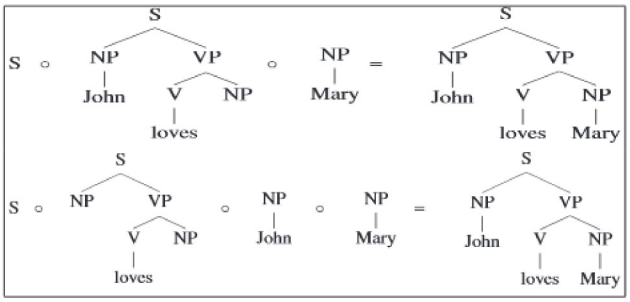

Fig. 2. Two different derivations of the same parse

[5]. An HBSG generates every parse-tree through a unique derivation involving rewrite production that can be considered, to a large extent, disjoint events. This allows for simpler estimation procedures and more efficient parsing algorithms than is possible for current DOP models.

This paper addresses the DOP parameter estimation from a different angle than preceding work. Crucially, we observe that the space of subtrees of a DOP model does not merely constitute a set of disjoint events, but that it constitutes a hierarchical space. This space is structured by a partial order between the different derivations of the same subtree; the more independence assumptions a derivation involves, the lower it is in this hierarchy, just like different n-gram orders of the same word string. This partial order between a subtree and its derivations is characterized by the relation of "backoff", defined in the sequel. Subsequently, a DOP model can be viewed as an interpolation of different orders of derivations: a subtree, the derivations of that subtree obtained by one, two, ... independence assumptions between smaller subtrees. Based on this observation we suggest to combine the different derivations through *backoff*, rather than interpolation. This view leads to a simple, yet powerful recursive estimation procedure. The new DOP model, Backoff DOP, leads to improved parsing results. We report on experiments that show a 10-15% error reduction on a treebank on which the original DOP model already achieves excellent results.

2 The DOP Model

Like other treebank models, DOP extracts a finite set of rewrite productions, called *subtrees*, from the training treebank together with probability estimates. A connected subgraph of a treebank tree t is called a *subtree* iff it consists of one or more context-free productions[2] from t. Following [2], the set of rewrite productions of DOP consists of *all* the subtrees of the treebank trees. Figure 3 exemplifies the set of subtrees extracted from the treebank of Figure 1.

[2] Note that a non-leaf node labeled p in tree t dominating a sequence of nodes labeled $c_1, \cdots, c_n$ consists of a graph that represents the context-free production: $p \rightarrow c_1 \cdots c_n$.

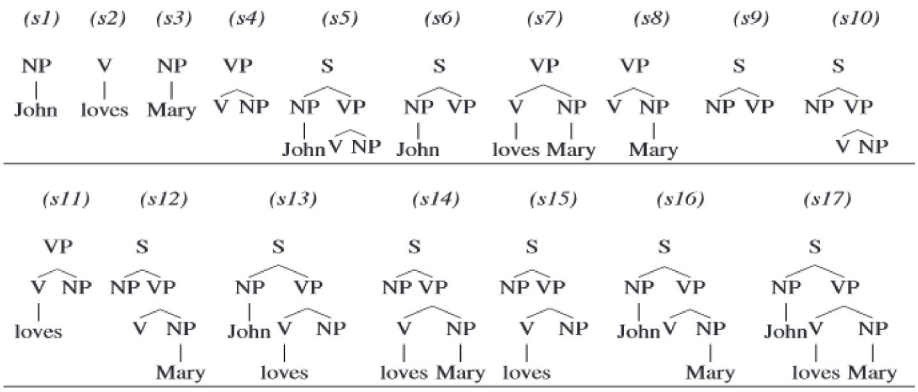

Fig. 3. The subtrees of the treebank in Figure 1

DOP employs the set of subtrees in a Stochastic Tree-Substitution Grammar (STSG): an TSG is a rewrite system similar to Context-Free Grammars (CFGs), with the difference that TSG productions are subtrees of arbitrary depth[3].

A TSG derivation proceeds by combining subtrees using the substitution operation $\circ$ starting from the start symbol S of the TSG. In contrast with CFG derivations, multiple TSG derivations may generate the same parse[4]. For example, the parse in Figure 1 can be derived at least in two different ways as shown in Figure 2. In this sense, the DOP model deviates from other contemporary models, e.g. [8,9], that belong to the so-called History-Based Stochastic Grammar (HBSG) family [5].

An Stochastic TSG (STSG) is a TSG extended with a probability mass function P over the set of subtrees: the probability of subtree t, that has root label $\mathtt{R}_t$, is given by $P(t\ \mathtt{R}_t)$, i.e. for every nonterminal A: $\sum_{\{t|\mathtt{R}_t=A\}} P(t\quad A) = 1$. Given a probability function P, the probability of a derivation $D = S \circ t_1 \circ \cdots \circ t_n$ is defined by $P(D \mid S) = \prod_{i=1}^{n} P(t_i\ \mathtt{R}_{t_i})$. The probability of a parse is defined by the sum of the probabilities of all derivations in the STSG that generate that parse.

When parsing an input sentence U under a DOP model, the preferred parse T is the Most Probable Parse (MPP) for that sentence: $\arg\max_T P(T\ U)$. However, the problem of computing the MPP is known to be intractable [10]. In contrast, the calculation of the Most Probable Derivation (MPD) D for the input sentence U i.e., $\arg\max_D P(D\ U)$, can be done in time cubic in sentence length.

In this paper we address another difficulty that arises from the property of the multiple derivations per parse with regard to the DOP model: how to estimate the model parameters (i.e. subtree probabilities) from a treebank.

[3] The depth of a tree is the number of edges along the longest path from the root to a leaf node.

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

3 Existing DOP Estimators

All three existing DOP estimators are biased, either by giving too much probability mass to large/small subtrees or by overfitting the training data.

(1) *Subtree relative Frequency:* The first instantiation of a DOP model is due to [11] and is referred to as DOP_{rf}. In this model, the probability estimates of subtrees extracted from a treebank are given by a relative frequency estimator. Let $f(t)$ represent the number of times t occurred in the bag of subtrees extracted from the treebank. Then the probability of t in DOP_{rf} is estimated as: $P_{rf}(t \mid \mathrm{R}_t) = \frac{f(t)}{\sum_{t':\mathrm{R}_{t'}=\mathrm{R}_t} f(t')}$.

Using heuristics to limit the unwanted biases, the model achieved 89.7% in labelled recall and precision on the Wall Street Journal treebank [1]. Despite good performance, DOP_{rf} estimator has been shown to be biased and inconsistent [4]. As argued in [3], DOP_{rf} overestimates the probability of large subtrees. Furthermore, DOP_{rf}'s good performance can be attributed to limitations on the set of subtrees extracted from the treebank (e.g. subtree depth upper bounds). These constraints reduce the model's bias, leading to improved performance.

(2) *Bonnema s Estimator:* In [3], an alternative estimator for DOP is proposed. It assumes that every treebank parse represents a *uniform distribution* over all possible derivations that generated that parse in the model. Thus, the probability of a subtree t is estimated by taking the relative frequency of t along with the fraction of derivations of the treebank parses in which t participates. This leads to the following estimate: $P(t \mid \mathrm{R}_t) = 2^{-N(t)}\ P_{rf}(t \mid \mathrm{R}_t)$, where $N(t)$ is the number of non-root nonterminal nodes of subtree t and P_{rf} is the original DOP_{rf}'s relative frequency estimator. The estimator defines a new DOP model which we refer to as DOP_{Bon} model. Next we show that the DOP_{Bon} estimator is biased towards smaller subtrees.

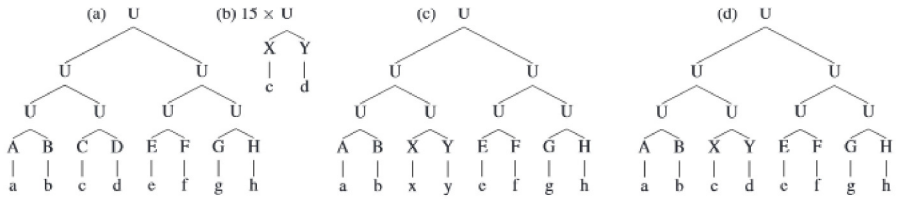

Fig. 4. /Example illustrating DOP_{Bon}'s bias towards small subtree fragments. Given the training treebank (a–c), the correct parse to *abcdefgh* should be (a). DOP_{Bon}model chooses (d) instead

Consider the treebank in Figure 4(a–c) with 17 subtrees. According to the treebank, the correct analysis for string *abcdefgh* should be the one in Figure 4(a). DOP_{Bon}, however, prefers the parse in Figure 4(d). In other words, a derivation that was actually seen in the treebank (i.e. the parse tree yielding *abcdefgh* in Figure 4(a)) becomes less likely than a newly constructed parse involving the subtree in Figure 4(b)!

(3) *Maximum-Likelihood:* One might say that DOP_{rf} estimator is biased because it is not a Maximum-Likelihood (ML) estimator. This is in fact the approach taken in [12], where the Inside-Outside algorithm is used for estimation of DOP model parameters from a treebank under the assumption that the model has a hidden element (the derivations that generated the parses of the treebank). However, as [3] pointed out, ML for DOP results in a model that overfits the treebank. We exemplify this next.

Let be given a treebank with trees τ_1, τ_2, both having the same root label X. The ML probability assignment to the subtrees extracted from this treebank is given as follows: for τ_1 and τ_2, $P(\tau_1\ X) = P(\tau_2\ X) = 1/2$; for all other X-rooted subtrees t, $P(t|X) = 0$. In other words, probability zero is given to all parses not present in the treebank, resulting in a model that overfits the data and has no generalization power.

4 A New Estimator for DOP

In this section we develop a completely different approach to parameter estimation for DOP than earlier work. Consider the common situation where a subtree[5] t is equal to a tree generated by a derivation $t_1 \circ \cdots \circ t_n$ involving multiple subtrees $t_1 \cdots t_n$. For example, subtree *s17* (Figure 3) can be constructed by different derivations such as *(s16 ∘ s2)*, *(s14 ∘ s1)* and *(s15 ∘ s1 ∘ s3)*. We will refer to subtrees that can be constructed from derivations involving other subtrees with the term *complex subtrees*.

For every complex subtree t, we restrict[6] our attention only to the derivations involving pairs of subtrees; in other words, we focus on subtree t such that there exist subtrees t_1 and t_2 such that $t = (t_1 \circ t_2)$. In DOP, the probability of t is given by $P(t\ \mathtt{R}_t)$. In contrast, the derivation probability is defined by $P(t_1 \circ t_2|\mathtt{R}_{t_1}) = P(t_1|\mathtt{R}_{t_1})P(t_2|\mathtt{R}_{t_2})$. However, according to the chain rule (note that $\mathtt{R}_t = \mathtt{R}_{t_1}$)

$$P(t_1 \circ t_2|\mathtt{R}_{t_1}) = P(t_1\ \mathtt{R}_{t_1})P(t_2\ t_1, \mathtt{R}_{t_1})$$

Therefore, the derivation $t_1 \circ t_2$ embodies an independence assumption realized by the approximation[7]: $P(t_2\ t_1) \approx P(t_2|\mathtt{R}_{t_2})$. This approximation involves a so-called *backoff*, i.e. a weakening of the conditioning context from $P(t_2|t_1)$ to $P(t_2\ \mathtt{R}_{t_2})$. Hence, we will say that the derivation $t_1 \circ t_2$ constitutes *a backoff* of subtree t and we will write $(t \geq_{bfk} t_1 \circ t_2)$ to express this fact.

[5] According to the definitions in section 2, the term "subtree" is reserved for the tree-structures that DOP extracts from the treebank.

[6] Because DOP takes all subtrees of the treebank, if complex subtree t has a derivation $t_1 \circ t_2 \circ \cdots \circ t_n$, then the tree resulting from $t_1 \circ t_2$ is a complex subtree also. For example, in Figure 3, $s17$ can be derived through *(s15 ∘ s1 ∘ s3)*; $s15 \circ s1$ generates subtree $s13$. Hence, derivations of t that involve more than two subtrees can be separarted into (sub)derivations that involve pairs of subtrees, each leading to a complex subtree. Therefore, for any complex subtree t, we may restrict our attention to derivations involving only pairs of subtrees i.e., $t = t_1 \circ t_2$.

[7] Note that $\mathtt{R}_{t_2}$ is part of t_1 (the label of the substitution site).

The backoff relation $\geq_{bfk}$ between a subtree and a pair of other subtrees allows for a partial order between the derivations of the subtrees extracted from a treebank. A graphical representation of this partial order is a directed acyclic graph which consists of a node for each pair of subtrees t_i, t_j that constitute a derivation of another complex subtree. A directed edge points from a subtree t_i in a node[8] to another node containing a pair of subtrees $\langle t_j, t_k \rangle$ iff $t_i \geq_{bfk} t_j \circ t_k$. We refer to this graph as the *backoff graph*. A portion of the backoff graph for the subtrees of Figure 3 is shown in figure 5 (where $s0$ stands for a subtree consisting of a single node labeled S – the start symbol).

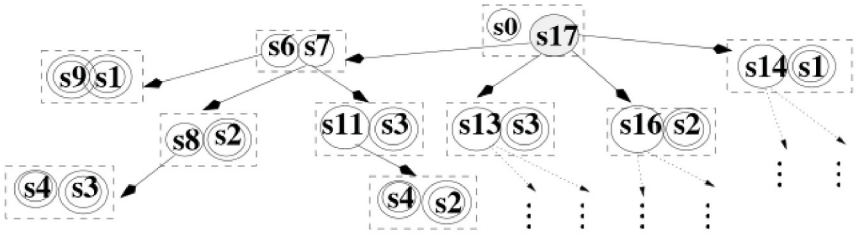

Fig. 5. A portion of the backoff graph for the subtrees in Figure 3

We distinguish two sets of subtrees: initial and atomic. Initial subtrees do not participate in a backoff derivation of any other subtree. In Figure 3, subtree $s17$ is the only initial subtree. Atomic subtrees are subtrees for which there are no backoffs. In Figure 3, these are subtrees of depth one (double circled in the backoff graph).

In the DOP model (under any estimation procedure discussed in section 3), the probablity of a parse-tree is defined as the sum of the probabilities of all derivations that generate this parse-tree. This means that DOP interpolates, linearly and with uniform weights, derivations involving subtrees from different levels of the backoff graph; this is similar to the way Hidden Markov Models interpolate different Markov orders over, e.g. words, for calculating sentence probability. Hence, we will refer to the different levels of subtrees in the backoff graph as the *Markov orders*.

Backoff DOP: Crucially, the partial order over the subtrees, embodied in the backoff graph, can be exploited for turning DOP into a "backedoff model" as follows. A subtree is generated by a sequence of derivations *ordered by the backoff relation*. This is in sharp contrast with existing DOP models that consider the different derivations leading to the same subtree as a set of *disjoint* events. Next, after we review smoothing and Katz Backoff, we present the estimation procedure that accompanies this new realization of DOP as a recursive backoff over the different Markov orders.

[8] In a pair $\langle t_h, t_i \rangle$ or $\langle t_i, t_h \rangle$ that constitutes a node.

Estimation vs. Smoothing: It is common to *smooth* a probability distribution $P(t\ X,Y)$ by a backoff distribution e.g. $P(t\ X)$. The smoothing of $P(t\ X,Y)$ aims at dealing with the problem of sparse-data (whenever the probability $P(t\ X,Y)$ is zero). $P(t\ X)$ can be used as an approximation of $P(t\ X,Y)$ under the assumption that t and Y are independent. Smoothing, then, aims at enlarging the space of non-zero events in the distribution $P(t\ X,Y)$. Hence, the goal of smoothing differs from our goal. While smoothing aims at filling the zero gaps in a distribution, our goal is to estimate the distribution (a priori to smoothing it). Despite these differences, we employ a backoff method for parameter estimation by *redistributing probability mass among DOP model subtrees.*

Katz Backoff: [6,7] is a smoothing technique based on the discounting method of Good-Turing (GT) [13,7]. Given a higher order distribution $P(t|X,Y)$, Katz backoff employs the GT formula for discounting from this distribution leading to $P_{GT}(t|X,Y)$. The probability mass that was discounted $(1-\sum_t P_{GT}(t|X,Y))$ is distributed over the lower order distribution $P(t\ X)$.

Estimation by Backoff: We assume initial probability estimates P_f based on frequency counts, e.g. as in DOP_{rf} or DOP_{Bon}. The present backoff estimation procedure operates iteratively, top-down over the backoff graph, starting with the initial and moving down towards the atomic subtrees. In essense this procedure *transfers, stepwisely, probability mass* from complex subtrees to their backoffs.

Let P^c represent the current probability estimate resulting from i previous steps of re-estimation (initially, for $i=0$, $P^0 := P_f$). After i steps, the edges of the backoff graph lead to the *current layer* of nodes. For every t, a subtree in a node from the current layer in the backoff graph, an edge e outgoing from t stands for the relation $(t \geq_{bfk} t_1 \circ t_2)$, where $\langle t_1, t_2\rangle$ is the node at the other end of edge e. We know that $P^c(t_2\ t_1)$ is backedoff to $P^c(t_2\ \mathtt{R}_{t_2})$ since $P^c(t|\mathtt{R}_t) = P^c(t_1\ \mathtt{R}_{t_1})P^c(t_2\ t_1)$ and $P^c(t_1 \circ t_2) = P^c(t_1\ \mathtt{R}_{t_1})P^c(t_2\ \mathtt{R}_{t_2})$. We adapt the Katz method to estimate the Backoff DOP probability P_{bo} as follows:

$$P_{bo}(t_2|t_1) = \begin{cases} P^c_{GT}(t_2\ t_1) \ +\ \alpha(t_1)\ P_f(t_2\ \mathtt{R}_{t_2}) & [\mathtt{P^c(t_2\ t_1) > 0}] \\ \alpha(t_1)\ P_f(t_2\ \mathtt{R}_{t_2}) & \mathtt{otherwise} \end{cases}$$

where $\alpha(t_1)$ is a normalization factor that guarantees that the sum of the probabilities of subtrees with the same root label is one. Simple arithmetic leads to the following formula: $\alpha(t_1) = 1 - \sum_{t_2:f(t_1,t_2)>0} P^c_{GT}(t_2\ t_1)$. Using the above estimate of $P_{bo}(t_2|t_1)$, the other backoff estimates are calculated as follows:

$$P_{bo}(t\ \mathtt{R}_t) := P_f(t_1\ \mathtt{R}_{t_1})\ P^c_{GT}(t_2|t_1) \qquad P_{bo}(t_1|\mathtt{R}_{t_1}) := (\ 1+\alpha(t_1)\ )\ P_f(t_1\ \mathtt{R}_{t_1})$$

Before the next step $i+1$ takes place over the next layer in the backoff graph, the current probabilities are updated as follows: $P^{i+1}(t_1|\mathtt{R}_{t_1}) := P_{bo}(t_1|\mathtt{R}_{t_1})$.

Note that P_{bo} is a proper distribution in the sense that for all nonterminals A: $\sum_t P(t\ A) = 1$. This is guaranteed by the redistribution of the reserved probability mass at every step of the procedure over the layers of the backoff graph. Furthermore, we note that the present method is *not* a smoothing method since it applies Katz Backoff for redistributing probability mass *only* among

subtrees that *did occur* in the treebank. The present method does not address probability estimation for unknown/unseen events.

Current Implementation: The number of subtrees extracted from a tree-bank is extremely large. In this paper, we choose to apply the Katz backoff only to $t \geq_{bfk} t_1 \circ t_2$ iff t_2 is a lexical subtree i.e., $t_2 = X \rightarrow w$ where X is a Part of Speech (PoS) tag and w a word. Our choice has to do with the importance of lexicalized subtrees and the overestimation that accompanies their relative frequency. All experiments reported here pertain to applying the backoff estimation procedure to this limited set of subtrees (while the probabilities of all other subtrees are left untouched).

5 Empirical Results

OVIS Corpus and Evaluation Metrics: The experiments were carried out on the OVIS corpus, a Dutch language, speech-based, dialogue system that provides railway information to human users over ordinary telephone lines [14]. The corpus contains 10,049 syntactically and semantically annotated utterances which are answers given by users to the system's questions (e.g. "From where to where do you want to travel?"). Utterances are annotated by a phrase-structure scheme with syntactic+semantic labels.

The corpus was randomly split into two sets: i) a training set with 9,049 trees; ii) a test set with 1,000 trees. The experiments were carried out using the same train/test split, unless stated otherwise. We report results for sentences that are at least two-word long (as 1 word sentences are easy). Without 1-word sentences, the average sentence length is 4.6 words/sentence.

Three accuracy measures were employed: exact match and recall/precision (F-score) of labeled bracketing [15]. Furthermore, we compate models using the *error reduction ratio*, the ratio between the percent point improvement of model 1 over model 2 normalized by the global error of model 2.

Subtree space was reduced by means of four upper bounds on their shape: 1) depth (□), 2) number of lexical items (□), 3) number of substitution sites (□) and 4) number of consecutive lexical items (□). Most Probable Derivation (**MPD**) and Most Probable Parse (**MPP**) were used as the maximization entities to select the preferred parse[9].

Naming convention: We tested the new estimator under two different counting strategies: DOP_{rf} and DOP_{Bon}. The following naming convention was used: □ □ □$_{□□}$ (as in [2]), □ □ □$_{□\,□□}$ (as in [3]), □ □ -□ □ □$_{□□}$ (backoff estimator applied to DOP_{rf} frequencies), and □ □ -□ □ □$_{□\,□□}$ (backoff estimator applied to DOP_{Bon} frequencies).

Accuracy vs. Depth Upper Bound: Figure 6 shows exact match results as a function of subtree depth upper bound. The subtrees were restricted to at most 2

[9] A complete specification of the algorithms for extracting the MPD and MPP can be found in [16,17].

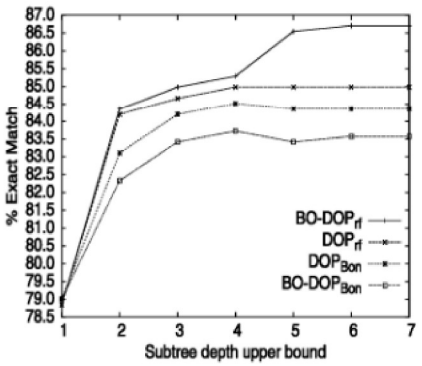

Fig. 6. Exact match as a function of subtree depth upper bound ($l2n4$, MPD)

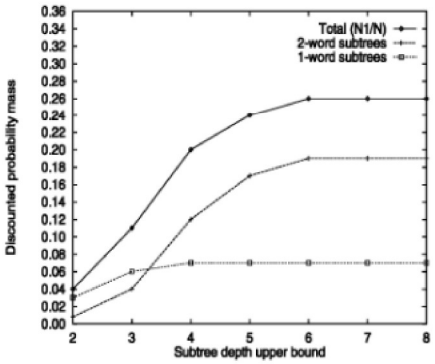

Fig. 7. Probability mass discounted from subtrees as a function of depth upper bound ($l2n4$)

words and 4 substitution nodes ($l2n4$). This yields small subtree spaces: for depth 7, this corresponds to 172,050 subtrees. For all depth upper bounds, $BF\text{-}DOP_{rf}$ achieved the best results followed by DOP_{rf}, DOP_{Bon} and $BF\text{-}DOP_{Bon}$. $BF\text{-}DOP_{rf}$ improved on DOP_{rf} by 1.72 percent points at depth 6, an increase of 2.02%. This corresponds to an error reduction of 11.4%. When compared to DOP_{Bon}, error reduction rose to about 15%. F-score results followed the same pattern. At depth 6, $BF\text{-}DOP_{rf}$ reached 95.33%; DOP_{rf}, 94.73%; DOP_{Bon}, 94.5% and $BF\text{-}DOP_{Bon}$, 94.33%. Error reduction of $BF\text{-}DOP_{rf}$ with respect to DOP_{rf} reached 11.3% and with respect to DOP_{Bon}, 15.7%.

The good perfomance of $BF\text{-}DOP_{rf}$ over DOP_{rf} may be explained by its reduced bias towards large subtrees. $BF\text{-}DOP_{Bon}$'s poor perfomance, on the other hand, is a result of increasing even further DOP_{Bon}'s bias towards smaller subtrees. The lesson here is: if one has to choose between biased estimators, choose the one favoring larger subtrees; they are able to capture more linguistically relevant dependencies.

Probability Mass Transfer: Figure 7 shows discounted probability mass as a function of subtree depth upper bound. The probability mass discounted from 2-word subtrees is bigger than the mass discounted from 1-word subtrees[10]. This happens because the number of hapax legomena (subtrees that occur just once) tends to increase for higher d and l upper bounds, since more large subtrees with rare word combinations are allowed into the distribution. More hapax legomena results in higher discounting rates according to the Good-Turing method. Thus, the probability mass discounted from n-word subtrees is, in general, bigger than the mass discounted from (n-1)-subtrees. Consequently, the magnitude of the probability transfer across Markov orders gradually decreases as the recursive estimation procedure approaches atomic subtrees. The property of decreasing

[10] n-word subtrees are subtrees having exactly n words in their leaf nodes.

discounts avoids the pitfall of overestimating small subtrees (c.f. DOP_{Bon}) and, at the same time, reduces the overestimation of large subtrees (c.f. DOP_{rf}).

Most Probable Parse and Derivation: The next set of experiments used distributions obtained with parameters l7n2L3. These settings allow for testing the models quickly, since they generate relatively small subtree spaces (133,308 subtrees for depth 7). Longer cascade effects can also be observed, since the models can backoff from 7-word subtrees to 0-word ones. Figures 8 and 9 show exact

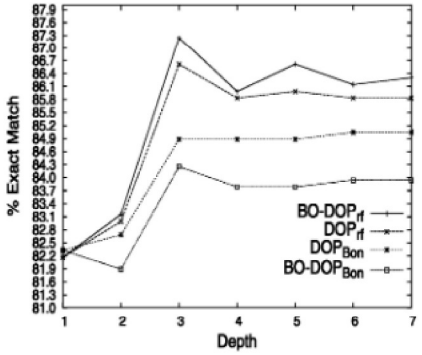

Fig. 8. Exact match as a function of depth upper bound (l7n2L3,MPD)

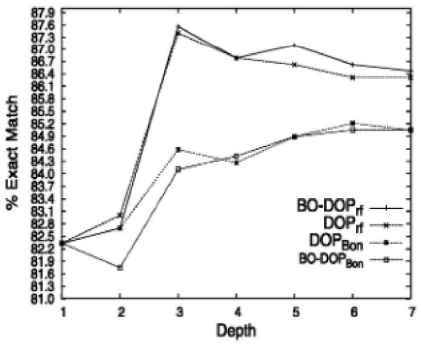

Fig. 9. Match as a function of depth upper bound (l7n2L3, MPP, Monte Carlo sample size: 5,000)

match results as a function of depth upper bound and maximization entity. Note that they follow the same pattern observed in the results with l2n4. Unlike those, however, maximum accuracy is achieved here at depth 3, not 6.

MPP has better performance than MPD. This shift is stronger for DOP_{rf} and $BF\text{-}DOP_{Bon}$. One possible explanation is that MPP allows for recovering part of the joint dependencies lost by the independence assumption underlying MPD. Moreover, contrary to MPP, MPD assumes that the probability of a parse is concentrated in a single derivation, which might lead to wrong results.

Consistency Across Splits: To test whether the results above are due to some random property of the train/test split, experiments were carried out with fixed parameters l7n2L3d3 and varying train/test splits. The number of trees was kept constant: 9,049 for training; 1,000 for testing. Depth 3 was chosen because most models achieved their best results with this setting, which might indicate fluctuation. Four experiments were carried out. Experiment 1 (Exp1) refers to the split used so far, Experiments 2, 3 and 4 (Exp2, Exp3, Exp4) refer to different splits obtained through random drawings from the OVIS corpus. Note that different splits yield distinct models with different generative powers. Therefore, performance can greatly vary.

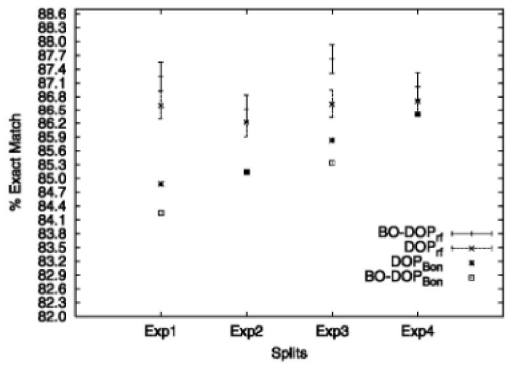

Fig. 10. Exact match for fixed depth 3 as a function of training/test set splits (l7n2d3L3,MPD)

Figure 10 shows that the pattern previously observed, with $BF\text{-}DOP_{rf}$ achieving the best results and $BF\text{-}DOP_{Bon}$ the worst, is conserved across the splits. The improvement, although persistent, is not statistically significant. $BF\text{-}DOP_{rf}$'s performance mean of 87.10% is not significantly different from DOP_{rf}'s mean of 86.54%, with 95% confidence according to the t-test (interval: $\pm 0.3097\%$).

Exp3 was the only one that reached statistical significance. It is important to emphasize that the wide range of the confidence interval is due to the small sample size. Definitive conclusions can only be drawn once a larger number of experiments is carried out. Moreover, these results refer to a single 'cut' in the parameter space and it is possible that different parameter combinations will result in significant differences. In any case, these results do not contradict the relative ranking between $BF\text{-}DOP_{rf}$ and DOP_{rf}.

6 Conclusions

The main point of this paper is that the DOP parameters constitute a hierarchically structured space of highly correlated events. We presented a novel estimator for the DOP model based on this observation by expressing the correlations in terms of backoff. We provided empirical evidence for the improved performance of this estimator over existing estimators.

We think that the hierarchical structuring of the space of DOP parameters can be exploited within a Maximum-Likelihood estimation procedure. The space structure can be seen to express some parameters as functions of other parameters.

Future work will address (1) formal aspects of the new estimator (bias and inconsistency questions), (2) a Maximum-Likelihood variant for DOP that incorporates the observations discussed in this paper, and (3) further experiments on larger treebanks, and less constrained DOP models.

References

1. Bod, R.: What is the minimal set of fragments that achieves maximal parse accuracy? In: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'2001). (2001)

2. Bod, R.: Enriching Linguistics with Statistics: Performance models of Natural Language. PhD dissertation. ILLC dissertation series 1995-14, University of Amsterdam (1995)
3. Bonnema, R., Buying, P., Scha, R.: A new probability model for data oriented parsing. In Dekker, P., ed.: Proceedings of the Twelfth Amsterdam Colloquium. ILLC/Department of Philosophy, University of Amsterdam, Amsterdam (1999) 85–90
4. Johnson, M.: The DOP estimation method is biased and inconsistent. Computational Linguistics **28(1)** (2002) 71–76
5. Black, E., Jelinek, F., Lafferty, J., Magerman, D., Mercer, R., Roukos, S.: Towards History-based Grammars: Using Richer Models for Probabilistic Parsing. In: Proceedings of the 31st Annual Meeting of the ACL (□□□'93), Columbus, Ohio (1993)
6. Katz, S.: Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Transactions on Acoustics, Speech and Signal Processing (ASSP) **35(3)** (1987) 400–401
7. Chen, S., Goodman, J.: An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University (1998)
8. Chelba, C., Jelinek, F.: Exploiting syntactic structure for language modeling. In Boitet, C., Whitelock, P., eds.: Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics, San Francisco, California, Morgan Kaufmann Publishers (1998) 225–231
9. Charniak, E.: A maximum entropy inspired parser. In: Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-00), Seattle, Washington, USA (2000) 132–139
10. Sima'an, K.: Computational complexity of probabilistic disambiguation. Grammars **5(2)** (2002) 125–151
11. Bod, R.: A computational model of language performance: Data Oriented Parsing. In: Proceedings of the 14th International Conference on Computational Linguistics (COLING'92), Nantes (1992)
12. Bod, R.: Combining semantic and syntactic structure for language modeling. In: Proceedings ICSLP-2000. (2000)
13. Good, I.: The population frequencies of species and the estimation of population parameters. Biometrika **40** (1953) 237–264
14. Scha, R., Bonnema, R., Bod, R., Sima'an, K.: Disambiguation and Interpretation of Wordgraphs using Data Oriented Parsing. Technical Report #31, Netherlands Organization for Scientific Research (NWO), Priority Programme Language and Speech Technology, *http* : *//grid.let.rug.nl* : 4321/ (1996)
15. Black et al., E.: A procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars. In: Proceedings of the February 1991 DARPA Speech and Natural Language Workshop, San Mateo, CA., Morgan Kaufman (1991) 306–311
16. Sima'an, K.: Learning Efficient Disambiguation. PhD dissertation (University of Utrecht). ILLC dissertation series 1999-02, University of Amsterdam, Amsterdam (1999)
17. Bod, R.: Beyond Grammar: An Experience-Based Theory of Language. CSLI Publications, California (1998)