Is the End of Supervised Parsing in Sight?

Rens Bod
School of Computer Science
University of St Andrews, ILLC, University of Amsterdam
rb@cs.st-and.ac.uk

Abstract

How far can we get with unsupervised parsing if we make our training corpus several orders of magnitude larger than has hitherto be attempted? We present a new algorithm for unsupervised parsing using an all-subtrees model, termed U-DOP*, which parses directly with packed forests of all binary trees. We train both on Penn's WSJ data and on the (much larger) NANC corpus, showing that U-DOP* outperforms a treebank-PCFG on the standard WSJ test set. While U-DOP* performs worse than state-of-the-art supervised parsers on hand-annotated sentences, we show that the model outperforms supervised parsers when evaluated as a language model in syntax-based machine translation on Europarl. We argue that supervised parsers miss the fluidity between constituents and non-constituents and that in the field of syntax-based language modeling the end of supervised parsing has come in sight.

1 Introduction

A major challenge in natural language parsing is the unsupervised induction of syntactic structure. While most parsing methods are currently supervised or semi-supervised (McClosky et al. 2006; Henderson 2004; Steedman et al. 2003), they depend on hand-annotated data which are difficult to come by and which exist only for a few languages. Unsupervised parsing methods are becoming increasingly important since they operate with raw, unlabeled data of which unlimited quantities are available.

There has been a resurgence of interest in unsupervised parsing during the last few years. Where van Zaanen (2000) and Clark (2001) induced unlabeled phrase structure for small domains like the ATIS, obtaining around 40% unlabeled f-score, Klein and Manning (2002) report 71.1% f-score on Penn WSJ part-of-speech strings ≤ 10 words (WSJ10) using a constituent-context model called CCM. Klein and Manning (2004) further show that a hybrid approach which combines constituency and dependency models, yields 77.6% f-score on WSJ10.

While Klein and Manning's approach may be described as an "all-substrings" approach to unsupervised parsing, an even richer model consists of an "all-subtrees" approach to unsupervised parsing, called U-DOP (Bod 2006). U-DOP initially assigns all unlabeled binary trees to a training set, efficiently stored in a packed forest, and next trains subtrees thereof on a held-out corpus, either by taking their relative frequencies, or by iteratively training the subtree parameters using the EM algorithm (referred to as "UML-DOP"). The main advantage of an all-subtrees approach seems to be the direct inclusion of *discontiguous* context that is not captured by (linear) substrings. Discontiguous context is important not only for learning structural dependencies but also for learning a variety of non-contiguous constructions such as *nearest … to…* or *take … by surprise*. Bod (2006) reports 82.9% unlabeled f-score on the same WSJ10 as used by Klein and Manning (2002, 2004). Unfortunately, his experiments heavily depend on a priori sampling of subtrees, and the model becomes highly inefficient if larger corpora are used or longer sentences are included.

In this paper we will also test an alternative model for unsupervised all-subtrees

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 400–407,
Prague, Czech Republic, June 2007. ©2007 Association for Computational Linguistics

parsing, termed U-DOP*, which is based on the DOP* estimator by Zollmann and Sima'an (2005), and which computes the shortest derivations for sentences from a held-out corpus using all subtrees from all trees from an extraction corpus. While we do not achieve as high an f-score as the UML-DOP model in Bod (2006), we will show that U-DOP* can operate without subtree sampling, and that the model can be trained on corpora that are two orders of magnitude larger than in Bod (2006). We will extend our experiments to 4 million sentences from the NANC corpus (Graff 1995), showing that an f-score of 70.7% can be obtained on the standard Penn WSJ test set by means of unsupervised parsing. Moreover, U-DOP* can be directly put to use in bootstrapping structures for concrete applications such as syntax-based machine translation and speech recognition. We show that U-DOP* outperforms the supervised DOP model if tested on the German-English Europarl corpus in a syntax-based MT system.

In the following, we first explain the DOP* estimator and discuss how it can be extended to unsupervised parsing. In section 3, we discuss how a PCFG reduction for supervised DOP can be applied to packed parse forests. In section 4, we will go into an experimental evaluation of U-DOP* on annotated corpora, while in section 5 we will evaluate U-DOP* on unlabeled corpora in an MT application.

2 From DOP* to U-DOP*

DOP* is a modification of the DOP model in Bod (1998) that results in a statistically consistent estimator and in an efficient training procedure (Zollmann and Sima'an 2005). DOP* uses the all-subtrees idea from DOP: given a treebank, take all subtrees, regardless of size, to form a stochastic tree-substitution grammar (STSG). Since a parse tree of a sentence may be generated by several (leftmost) derivations, the probability of a tree is the sum of the probabilities of the derivations producing that tree. The probability of a derivation is the product of the subtree probabilities. The original DOP model in Bod (1998) takes the occurrence frequencies of the subtrees in the trees normalized by their root frequencies as subtree parameters. While efficient algorithms have been developed for this DOP model by converting it into a PCFG reduction (Goodman 2003), DOP's estimator was shown to be inconsistent by Johnson (2002). That is, even with unlimited training data, DOP's estimator is not guaranteed to converge to the correct distribution.

Zollmann and Sima'an (2005) developed a statistically consistent estimator for DOP which is based on the assumption that maximizing the joint probability of the parses in a treebank can be approximated by maximizing the joint probability of their shortest derivations (i.e. the derivations consisting of the fewest subtrees). This assumption is in consonance with the principle of simplicity, but there are also empirical reasons for the shortest derivation assumption: in Bod (2003) and Hearne and Way (2006), it is shown that DOP models that select the preferred parse of a test sentence using the shortest derivation criterion perform very well.

On the basis of this shortest-derivation assumption, Zollmann and Sima'an come up with a model that uses held-out estimation: the training corpus is randomly split into two parts proportional to a fixed ratio: an *extraction corpus* EC and a *held-out* corpus HC. Applied to DOP, held-out estimation would mean to extract fragments from the trees in EC and to assign their weights such that the likelihood of HC is maximized. If we combine their estimation method with Goodman's reduction of DOP, Zollman and Sima'an's procedure operates as follows:

(1) Divide a treebank into an EC and HC
(2) Convert the subtrees from EC into a PCFG reduction
(3) Compute the shortest derivations for the sentences in HC (by simply assigning each subtree equal weight and applying Viterbi 1-best)
(4) From those shortest derivations, extract the subtrees and their relative frequencies in HC to form an STSG

Zollmann and Sima'an show that the resulting estimator is consistent. But equally important is the fact that this new DOP* model does not suffer from a decrease in parse accuracy if larger subtrees are included, whereas the original DOP model needs to be redressed by a correction factor to maintain this property (Bod 2003). Moreover, DOP*'s estimation procedure is very efficient, while the EM training procedure for UML-DOP proposed in Bod (2006) is particularly time consuming and can only operate by randomly sampling trees.

Given the advantages of DOP*, we will generalize this model in the current paper to *unsupervised* parsing. We will use the same all-subtrees methodology as in Bod (2006), but now by applying the efficient and consistent DOP*-based estimator. The resulting model, which we will call U-DOP*, roughly operates as follows:

(1) Divide a corpus into an EC and HC
(2) Assign all unlabeled binary trees to the sentences in EC, and store them in a shared parse forest
(3) Convert the subtrees from the parse forests into a compact PCFG reduction (see next section)
(4) Compute the shortest derivations for the sentences in HC (as in DOP*)
(5) From those shortest derivations, extract the subtrees and their relative frequencies in HC to form an STSG
(6) Use the STSG to compute the most probable parse trees for new test data by means of Viterbi *n*-best (see next section)

We will use this U-DOP* model to investigate our main research question: *how far can we get with unsupervised parsing if we make our training corpus several orders of magnitude larger than has hitherto be attempted?*

3 Converting shared parse forests into PCFG reductions

The main computational problem is how to deal with the immense number of subtrees in U-DOP*. There exists already an efficient *supervised* algorithm that parses a sentence by means of all subtrees from a treebank. This algorithm was extensively described in Goodman (2003) and converts a DOP-based STSG into a compact PCFG reduction that generates eight rules for each node in the treebank. The reduction is based on the following idea: every node in every treebank tree is assigned a unique number which is called its address. The notation $A@k$ denotes the node at address k where A is the nonterminal labeling that node. A new nonterminal is created for each node in the training data. This nonterminal is called A_k. Let a_j represent the number of subtrees headed by the node $A@j$, and let a represent the number of subtrees headed by nodes with nonterminal A, that is $a = \Sigma_j\ a_j$. Then there is a PCFG with the following property: for every subtree in the training corpus headed by A, the grammar will generate an isomorphic subderivation. For example, for a node ($A@j$ ($B@k$, $C@l$)), the following eight PCFG rules in figure 1 are generated, where the number following a rule is its weight.

$A_j \rightarrow BC$	$(1/a_j)$	$A \rightarrow BC$	$(1/a)$
$A_j \rightarrow B_kC$	(b_k/a_j)	$A \rightarrow B_kC$	(b_k/a)
$A_j \rightarrow BC_l$	(c_l/a_j)	$A \rightarrow BC_l$	(c_l/a)
$A_j \rightarrow B_kC_l$	(b_kc_l/a_j)	$A \rightarrow B_kC_l$	(b_kc_l/a)

Figure 1. PCFG reduction of supervised DOP

By simple induction it can be shown that this construction produces PCFG derivations isomorphic to DOP derivations (Goodman 2003: 130-133). The PCFG reduction is linear in the number of nodes in the corpus.

While Goodman's reduction method was developed for supervised DOP where each training sentence is annotated with exactly one tree, the method can be generalized to a corpus where each sentence is annotated with all possible binary trees (labeled with the generalized category X), as long as we represent these trees by a shared parse forest. A shared parse forest can be obtained by adding pointers from each node in the chart (or tabular diagram) to the nodes that caused it to be placed in the chart. Such a forest can be represented in cubic space and time (see Billot and Lang 1989). Then, instead of assigning a unique address to each node in each tree, as done by the PCFG reduction for supervised DOP, we now assign a unique address to each node in each parse forest for each sentence. However, the same node may be part of more than one tree. A shared parse forest is an AND-OR graph where AND-nodes correspond to the usual parse tree nodes, while OR-nodes correspond to distinct subtrees occurring in the same context. The total number of nodes is cubic in sentence length n. This means that there are $O(n^3)$ many nodes that receive a unique address as described above, to which next our PCFG reduction is applied. This is a huge reduction compared to Bod (2006) where the number of subtrees of all trees increases with the Catalan number, and only ad hoc sampling could make the method work.

Since U-DOP* computes the shortest derivations (in the training phase) by combining subtrees from unlabeled binary trees, the PCFG reduction in figure 1 can be represented as in figure 2, where *X* refers to the generalized category while *B* and *C* either refer to part-of-speech categories or are equivalent to *X*. The equal weights follow from the fact that the shortest derivation is equivalent to the most probable derivation if all subtrees are assigned equal probability (see Bod 2000; Goodman 2003).

$X_j \rightarrow BC$	1	$X \rightarrow BC$	0.5
$X_j \rightarrow B_kC$	1	$X \rightarrow B_kC$	0.5
$X_j \rightarrow BC_l$	1	$X \rightarrow BC_l$	0.5
$X_j \rightarrow B_kC_l$	1	$X \rightarrow B_kC_l$	0.5

Figure 2. PCFG reduction for U-DOP*

Once we have parsed HC with the shortest derivations by the PCFG reduction in figure 2, we extract the subtrees from HC to form an STSG. The number of subtrees in the shortest derivations is linear in the number of nodes (see Zollmann and Sima'an 2005, theorem 5.2). This means that U-DOP* results in an STSG which is much more succinct than previous DOP-based STSGs. Moreover, as in Bod (1998, 2000), we use an extension of Good-Turing to smooth the subtrees and to deal with 'unknown' subtrees.

Note that the direct conversion of parse forests into a PCFG reduction also allows us to efficiently implement the maximum likelihood extension of U-DOP known as UML-DOP (Bod 2006). This can be accomplished by training the PCFG reduction on the held-out corpus HC by means of the expectation-maximization algorithm, where the weights in figure 1 are taken as initial parameters. Both U-DOP*'s and UML-DOP's estimators are known to be statistically consistent. But while U-DOP*'s training phase merely consists of the computation of the shortest derivations and the extraction of subtrees, UML-DOP involves iterative training of the parameters.

Once we have extracted the STSG, we compute the most probable parse for new sentences by Viterbi *n*-best, summing up the probabilities of derivations resulting in the same tree (the exact computation of the most probable parse is NP hard – see Sima'an 1996). We have incorporated the technique by Huang and Chiang (2005) into our implementation which allows for efficient Viterbi *n*-best parsing.

4 Evaluation on hand-annotated corpora

To evaluate U-DOP* against UML-DOP and other unsupervised parsing models, we started out with three corpora that are also used in Klein and Manning (2002, 2004) and Bod (2006): Penn's WSJ10 which contains 7422 sentences ≤ 10 words after removing empty elements and punctuation, the German NEGRA10 corpus and the Chinese Treebank CTB10 both containing 2200+ sentences ≤ 10 words after removing punctuation. As with most other unsupervised parsing models, we train and test on p-o-s strings rather than on word strings. The extension to word strings is straightforward as there exist highly accurate unsupervised part-of-speech taggers (e.g. Schütze 1995) which can be directly combined with unsupervised parsers, but for the moment we will stick to p-o-s strings (we will come back to word strings in section 5). Each corpus was divided into 10 training/test set splits of 90%/10% (*n*-fold testing), and each training set was randomly divided into two equal parts, that serve as EC and HC and vice versa. We used the same evaluation metrics for unlabeled precision (UP) and unlabeled recall (UR) as in Klein and Manning (2002, 2004). The two metrics of UP and UR are combined by the unlabeled f-score F1 = 2*UP*UR/(UP+UR). All trees in the test set were binarized beforehand, in the same way as in Bod (2006).

For UML-DOP the decrease in cross-entropy became negligible after maximally 18 iterations. The training for U-DOP* consisted in the computation of the shortest derivations for the HC from which the subtrees and their relative frequencies were extracted. We used the technique in Bod (1998, 2000) to include 'unknown' subtrees. Table 1 shows the f-scores for U-DOP* and UML-DOP against the f-scores for U-DOP reported in Bod (2006), the CCM model in Klein and Manning (2002), the DMV dependency model in Klein and Manning (2004) and their combined model DMV+CCM.

Model	English (WSJ10)	German (NEGRA10)	Chinese (CTB10)
CCM	71.9	61.6	45.0
DMV	52.1	49.5	46.7
DMV+CCM	77.6	63.9	43.3
U-DOP	78.5	65.4	46.6
U-DOP*	77.9	63.8	42.8
UML-DOP	79.4	65.2	45.0

Table 1. F-scores of U-DOP* and UML-DOP compared to other models on the same data.

It should be kept in mind that an exact comparison can only be made between U-DOP* and UML-DOP in table 1, since these two models were tested on 90%/10% splits, while the other models were applied to the full WSJ10, NEGRA10 and CTB10 corpora. Table 1 shows that U-DOP* performs worse than UML-DOP in all cases, although the differences are small and was statistically significant only for WSJ10 using paired *t*-testing.

As explained above, the main advantage of U-DOP* over UML-DOP is that it works with a more succinct grammar extracted from the shortest derivations of HC. Table 2 shows the size of the grammar (number of rules or subtrees) of the two models for resp. Penn WSJ10, the entire Penn WSJ and the first 2 million sentences from the NANC (North American News Text) corpus which contains a total of approximately 24 million sentences from different news sources.

Model	Size of STSG for WSJ10	Size of STSG for Penn WSJ	Size of STSG for 2,000K NANC
U-DOP*	2.2×10^4	9.8×10^5	7.2×10^6
UML-DOP	1.5×10^6	8.1×10^7	5.8×10^9

Table 2. Grammar size of U-DOP* and UML-DOP for WSJ10 (7,7K sentences), WSJ (50K sentences) and the first 2,000K sentences from NANC.

Note that while U-DOP* is about 2 orders of magnitudes smaller than UML-DOP for the WSJ10, it is almost 3 orders of magnitudes smaller for the first 2 million sentences of the NANC corpus. Thus even if U-DOP* does not give the highest f-score in table 1, it is more apt to be trained on larger data sets. In fact, a well-known advantage of unsupervised methods over supervised methods is the availability of almost unlimited amounts of text. Table 2 indicates that U-DOP*'s grammar is still of manageable size even for text corpora that are (almost) two orders of magnitude larger than Penn's WSJ. The NANC corpus contains approximately 2 million WSJ sentences that do not overlap with Penn's WSJ and has been previously used by McClosky et al. (2006) in improving a supervised parser by self-training. In our experiments below we will start by mixing subsets from the NANC's WSJ data with Penn's WSJ data. Next, we will do the same with 2 million sentences from the LA Times in the NANC corpus, and finally we will mix all data together for inducing a U-DOP* model. From Penn's WSJ, we only use sections 2 to 21 for training (just as in supervised parsing) and section 23 (≤100 words) for testing, so as to compare our unsupervised results with some binarized supervised parsers.

The NANC data was first split into sentences by means of a simple discriminitive model. It was next p-o-s tagged with the the TnT tagger (Brants 2000) which was trained on the Penn Treebank such that the same tag set was used. Next, we added subsets of increasing size from the NANC p-o-s strings to the 40,000 Penn WSJ p-o-s strings. Each time the resulting corpus was split into two halfs and the shortest derivations were computed for one half by using the PCFG-reduction from the other half and vice versa. The resulting trees were used for extracting an STSG which in turn was used to parse section 23 of Penn's WSJ. Table 3 shows the results.

# sentences added	f-score by adding WSJ data	f-score by adding LA Times data
0 (baseline)	62.2	62.2
100k	64.7	63.0
250k	66.2	63.8
500k	67.9	64.1
1,000k	68.5	64.6
2,000k	69.0	64.9

Table 3. Results of U-DOP* on section 23 from Penn's WSJ by adding sentences from NANC's WSJ and NANC's LA Times

Table 3 indicates that there is a monotonous increase in f-score on the WSJ test set if NANC text is added to our training data in both cases, independent of whether the sentences come from the WSJ domain or the LA Times domain. Although the effect of adding LA Times data is weaker than adding WSJ data, it is noteworthy that the unsupervised induction of trees from the LA Times domain still improves the f-score even if the test data are from a different domain.

We also investigated the effect of adding the LA Times data to the total mix of Penn's WSJ and NANC's WSJ. Table 4 shows the results of this experiment, where the baseline of 0 sentences thus starts with the 2,040k sentences from the combined Penn-NANC WSJ data.

Sentences added from LA Times to Penn-NANC WSJ	**f-score by adding LA Times data**
0	69.0
100k	69.4
250k	69.9
500k	70.2
1,000k	70.4
2,000k	70.7

Table 4. Results of U-DOP* on section 23 from Penn's WSJ by mixing sentences from the combined Penn-NANC WSJ with additions from NANC's LA Times.

As seen in table 4, the f-score continues to increase even when adding LA Times data to the large combined set of Penn-NANC WSJ sentences. The highest f-score is obtained by adding 2,000k sentences, resulting in a total training set of 4,040k sentences. We believe that our result is quite promising for the future of unsupervised parsing.

In putting our best f-score in table 4 into perspective, it should be kept in mind that the gold standard trees from Penn-WSJ section 23 were binarized. It is well known that such a binarization has a negative effect on the f-score. Bod (2006) reports that an unbinarized treebank grammar achieves an average 72.3% f-score on WSJ sentences ≤ 40 words, while the binarized version achieves only 64.6% f-score. To compare U-DOP*'s results against some supervised parsers, we additionally evaluated a PCFG treebank grammar and the supervised DOP* parser using the same test set. For these supervised parsers, we employed the standard training set, i.e. Penn's WSJ sections 2-21, but only by taking the p-o-s strings as we did for our unsupervised U-DOP* model. Table 5 shows the results of this comparison.

Parser	**f-score**
U-DOP*	70.7
Binarized treebank PCFG	63.5
Binarized DOP*	80.3

Table 5. Comparison between the (best version of) U-DOP*, the supervised treebank PCFG and the supervised DOP* for section 23 of Penn's WSJ

As seen in table 5, U-DOP* outperforms the binarized treebank PCFG on the WSJ test set. While a similar result was obtained in Bod (2006), the absolute difference between unsupervised parsing and the treebank grammar was extremely small in Bod (2006): 1.8%, while the difference in table 5 is 7.2%, corresponding to 19.7% error reduction. Our f-score remains behind the supervised version of DOP* but the gap gets narrower as more training data is being added to U-DOP*.

5 Evaluation on unlabeled corpora in a practical application

Our experiments so far have shown that despite the addition of large amounts of unlabeled training data, U-DOP* is still outperformed by the supervised DOP* model when tested on hand-annotated corpora like the Penn Treebank. Yet it is well known that any evaluation on hand-annotated corpora unreasonably favors supervised parsers. There is thus a quest for designing an evaluation scheme that is independent of annotations. One way to go would be to compare supervised and unsupervised parsers as a syntax-based language model in a practical application such as machine translation (MT) or speech recognition.

In Bod (2007), we compared U-DOP* and DOP* in a syntax-based MT system known as Data-Oriented Translation or *DOT* (Poutsma 2000; Groves et al. 2004). The DOT model starts with a bilingual treebank where each tree pair constitutes an example translation and where translationally equivalent constituents are linked. Similar to DOP, the DOT model uses all linked subtree pairs from the bilingual treebank to form an STSG of linked subtrees, which are used to compute the most probable translation of a target sentence given a source sentence (see Hearne and Way 2006).

What we did in Bod (2007) is to let both DOP* and U-DOP* compute the best trees directly for the *word strings* in the German-English Europarl corpus (Koehn 2005), which contains about 750,000 sentence pairs. Differently from U-DOP*, DOP* needed to be trained on annotated data, for which we used respectively the Negra and the Penn treebank. Of course, it is well-known that a supervised parser's f-score decreases if it is transferred to another domain: for example, the (non-binarized) WSJ-trained DOP model in Bod (2003) decreases from around 91% to 85.5% f-score if tested on the Brown corpus. Yet, this score is still considerably higher than the accuracy obtained by the unsupervised U-DOP model, which achieves 67.6% unlabeled f-score on Brown sentences. Our main question of interest is in how far this difference in accuracy on hand-annotated corpora carries over when tested in the context of a concrete application like MT. This is not a trivial question, since U-DOP* learns 'constituents' for word sequences such as *Ich möchte* ("I would like to") and *There are* (Bod 2007), which are usually hand-annotated as non-constituents. While U-DOP* is punished for this 'incorrect' prediction if evaluated on the Penn Treebank, it may be rewarded for this prediction if evaluated in the context of machine translation using the Bleu score (Papineni et al. 2002). Thus similar to Chiang (2005), U-DOP can discover non-syntactic phrases, or simply "phrases", which are typically neglected by linguistically syntax-based MT systems. At the same time, U-DOP* can also learn discontiguous constituents that are neglected by phrase-based MT systems (Koehn et al. 2003).

In our experiments, we used both U-DOP* and DOP* to predict the best trees for the German-English Europarl corpus. Next, we assigned links between each two nodes in the respective trees for each sentence pair. For a 2,000 sentence test set from a different part of the Europarl corpus we computed the most probable target sentence (using Viterbi n best). The Bleu score was used to measure translation accuracy, calculated by the NIST script with its default settings. As a baseline we compared our results with the publicly available phrase-based system Pharaoh (Koehn et al. 2003), using the default feature set. Table 6 shows for each system the Bleu score together with a description of the productive units. 'U-DOT' refers to 'Unsupervised DOT' based on U-DOP*, while DOT is based on DOP*.

System	Productive Units	Bleu-score
U-DOT / U-DOP*	Constituents and Phrases	0.280
DOT / DOP*	Constituents only	0.221
Pharaoh	Phrases only	0.251

Table 6. Comparing U-DOP* and DOP* in syntax-based MT on the German-English Europarl corpus against the Pharaoh system.

The table shows that the unsupervised U-DOT model outperforms the supervised DOT model with 0.059. Using Zhang's significance tester (Zhang et al. 2004), it turns out that this difference is statistically significant (p < 0.001). Also the difference between U-DOT and the baseline Pharaoh is statistically significant (p < 0.008). Thus even if supervised parsers like DOP* outperform unsupervised parsers like U-DOP* on hand-parsed data with >10%, the same supervised parser is outperformed by the unsupervised parser if tested in an MT application. Evidently, U-DOP's capacity to capture both constituents and phrases pays off in a concrete application and shows the shortcomings of models that only allow for either constituents (such as linguistically syntax-based MT) or phrases (such as phrase-based MT). In Bod (2007) we also show that U-DOT obtains virtually the same Bleu score as Pharaoh after eliminating subtrees with discontiguous yields.

6 Conclusion: future of supervised parsing

In this paper we have shown that the accuracy of unsupervised parsing under U-DOP* continues to grow when enlarging the training set with additional data. However, except for the simple treebank PCFG, U-DOP* scores worse than supervised parsers if evaluated on hand-annotated data. At the same time U-DOP* significantly outperforms the supervised DOP* if evaluated in a practical application like MT. We argued that this can be explained by the fact that U-DOP learns both constituents and (non-syntactic) phrases while supervised parsers learn constituents only.

What should we learn from these results? We believe that parsing, when separated from a task-based application, is mainly an academic exercise. If we only want to mimick a treebank or implement a linguistically motivated grammar, then supervised, grammar-based parsers are preferred to unsupervised parsers. But if we want to improve a practical application with a syntax-based language model, then an unsupervised parser like U-DOP* might be superior.

The problem with most supervised (and semi-supervised) parsers is their rigid notion of constituent which excludes 'constituents' like the German *Ich möchte* or the French *Il y a*. Instead, it has become increasingly clear that the notion of constituent is a fluid which may sometimes be in agreement with traditional syntax, but which may just as well be in opposition to it. Any sequence of words can be a unit of combination, including non-contiguous word sequences like *closest* X *to* Y. A parser which does not allow for this fluidity may be of limited use as a language model. Since supervised parsers seem to stick to categorical notions of constituent, we believe that in the field of syntax-based language models the end of supervised parsing has come in sight.

Acknowledgements

Thanks to Willem Zuidema and three anonymous reviewers for useful comments and suggestions on the future of supervised parsing.

References

Billot, S. and B. Lang, 1989. The Structure of Shared Forests in Ambiguous Parsing. In *ACL 1989*.

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

Bod, R. Parsing with the Shortest Derivation. In *COLING 2000*, Saarbruecken.

Bod, R. 2003. An efficient implementation of a new DOP model. In *EACL 2003*, Budapest.

Bod, R. 2006. An All-Subtrees Approach to Unsupervised Parsing. In *ACL-COLING 2006*, Sydney.

Bod, R. 2007. Unsupervised Syntax-Based Machine Translation. Submitted for publication.

Brants, T. 2000. TnT - A Statistical Part-of-Speech Tagger. In *ANLP 2000*.

Chiang, D. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *ACL 2005*, Ann Arbor.

Clark, A. 2001. Unsupervised induction of stochastic context-free grammars using distributional clustering. In *CONLL 2001*.

Goodman, J. 2003. Efficient algorithms for the DOP model. In R. Bod, R. Scha and K. Sima'an (eds.). *Data-Oriented Parsing*, CSLI Publications.

Graff, D. 1995. *North American News Text Corpus*. Linguistic Data Consortium. LDC95T21.

Groves, D., M. Hearne and A. Way, 2004. Robust Sub-Sentential Alignment of Phrase-Structure Trees. In *COLING 2004*, Geneva.

Hearne, M and A. Way, 2006. Disambiguation Strategies for Data-Oriented Translation. *Proceedings of the 11th Conference of the European Association for Machine Translation*, Oslo.

Henderson, J. 2004. Discriminative training of a neural network statistical parser. In *ACL 2004,* Barcelona.

Huang, L. and D. Chiang 2005. Better *k*-best parsing. In *IWPT 2005*, Vancouver.

Johnson, M. 2002. The DOP estimation method is biased and inconsistent. *Computational Linguistics* 28, 71-76.

Klein, D. and C. Manning 2002. A general constituent-context model for improved grammar induction. In *ACL 2002*, Philadelphia.

Klein, D. and C. Manning 2004. Corpus-based induction of syntactic structure: models of dependency and constituency. *ACL 2004*, Barcelona.

Koehn, P., Och, F. J., and Marcu, D. 2003. Statistical phrase based translation. In *HLT-NAACL 2003*.

Koehn, P. 2005. Europarl: a parallel corpus for statistical machine translation. In *MT Summit 2005*.

McClosky, D., E. Charniak and M. Johnson 2006. Effective self-training for parsing. In *HLT-NAACL 2006*, New York.

Poutsma, A. 2000. Data-Oriented Translation. In *COLING 2000*, Saarbruecken.

Schütze, H. 1995. Distributional part-of-speech tagging. In *ACL 1995*, Dublin.

Sima'an, K. 1996. Computational complexity of probabilistic disambiguation by means of tree grammars. In *COLING 1996*, Copenhagen.

Steedman, M. M. Osborne, A. Sarkar, S. Clark, R. Hwa, J. Hockenmaier, P. Ruhlen, S. Baker, and J. Crim. 2003. Bootstrapping statistical parsers from small datasets. In *EACL 2003*, Budapest.

van Zaanen, M. 2000. ABL: Alignment-Based Learning. In *COLING 2000*, Saarbrücken.

Zhang, Y., S. Vogel and A. Waibel, 2004. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*.

Zollmann, A. and K. Sima'an 2005. A consistent and efficient estimator for data-oriented parsing. *Journal of Automata, Languages and Combinatorics*, Vol. 10 (2005) Number 2/3, 367-388.