constituency. It would be interesting to investigate an extension of U-DOP towards dependency parsing, which we will leave for future research. It is also noteworthy that U-DOP does not employ a separate class for non-constituents, so-called distituents, while CCM does. Thus good results can be obtained without keeping track of distituents but by simply assigning all binary trees to the strings and letting the DOP model decide which substrings are most likely to form constituents.

To give an idea of the constituents learned by U-DOP for the WSJ10, table 2 shows the 10 most frequently constituents in the trees induced by U-DOP together with the 10 actually most frequently occurring constituents in the WSJ10 and the 10 most frequently occurring part-of-speech sequences (bigrams) in the WSJ10.

Rank	Most frequent U-DOP constituents	Most Frequent WSJ10 constituents	Most frequent WSJ10 substrings
1	DT NN	DT NN	NNP NNP
2	NNP NNP	NNP NNP	DT NN
3	DT JJ NN	CD CD	JJ NN
4	IN DT NN	JJ NNS	IN DT
5	CD CD	DT JJ NN	NN IN
6	DT NNS	DT NNS	DT JJ
7	JJ NNS	JJ NN	JJ NNS
8	JJ NN	CD NN	NN NN
9	VBN IN	IN NN	CD CD
10	VBD NNS	IN DT NN	NN VBZ

Table 2. Most frequently learned constituents by U-DOP together with most frequently occurring constituents and p-o-s sequences (for WSJ10)

Note that there are no distituents among U-DOP's 10 most frequently learned constituents, whilst the third column shows that distituents such as IN DT or DT JJ occur very frequently as substrings in the WSJ10. This may be explained by the fact that (the constituent) DT NN occurs more frequently as a substring in the WSJ10 than (the distituent) IN DT, and therefore U-DOP's probability model will favor a covering subtree for IN DT NN which consists of a division into IN *X* and DT NN rather than into IN DT and *X* NN, other things being equal. The same kind reasoning can be made for a subtree for DT JJ NN where the constituent JJ NN occurs more frequently as a substring than the distituent DT JJ. Of course the situation is somewhat more complex in DOP's sum-of-products model, but our argument may illustrate why distituents like IN DT or DT JJ are not proposed among the most frequent constituents by U-DOP while larger constituents like IN DT NN and DT JJ NN are in fact proposed.

3.2 Testing U-DOP on held-out sets and longer sentences (up to 40 words)

We were also interested in U-DOP's performance on a held-out test set such that we could compare the model with a *supervised* PCFG treebank grammar trained and tested on the same data (S-PCFG). We started by testing U-DOP on 10 different 90%/10% splits of the WSJ10, where 90% was used for inducing the trees, and 10% to parse new sentences by subtrees from the binary trees from the training set (or actually a PCFG-reduction thereof). The supervised PCFG was right-binarized as in Klein and Manning (2005). The following table shows the results.

Model	UP	UR	F1
U-DOP	70.6	**88.1**	78.3
S-PCFG	**84.0**	79.8	**81.8**

Table 3. Average f-scores of U-DOP compared to a supervised PCFG (S-PCFG) on 10 different 90-10 splits of the WSJ10

Comparing table 1 with table 3, we see that on 10 held-out WSJ10 test sets U-DOP performs with an average f-score of 78.3% (SD=2.1%) only slightly worse than when using the entire WSJ10 corpus (78.5%). Next, note that U-DOP's results come near to the average performance of a binarized supervised PCFG which achieves 81.8% unlabeled f-score (SD=1.8%). U-DOP's unlabeled recall is even higher than that of the supervised PCFG. Moreover, according to paired *t*-testing, the differences in f-scores were *not* statistically significant. (If the PCFG was not post-binarized, its average f-score was 89.0%.)

As a final test case for this paper, we were interested in evaluating U-DOP on WSJ sentences ≤ 40 words, i.e. the WSJ40, which is with almost 50,000 sentences a much more challenging test case than the relatively small WSJ10. The main problem for U-DOP is the astronomically large number of possible binary trees for longer sentences, which therefore need to be even more heavily pruned than before.

We used a similar sampling heuristic as in section 2. We started by taking 100% of the trees for sentences ≤ 7 words. Next, for longer sentences we reduced this percentage with the relative increase of the Catalan number. This effectively means that we randomly selected the same number of trees for each sentence ≥ 8 words, which is 132 (i.e. the