Step 3: Compute the most probable parse tree for each WSJ10 string

While Goodman's reduction method allows for efficiently computing the most probable derivation for each sentence (i.e. the Viterbi parse), it does not allow for an efficient computation of (U-)DOP's most probable parse tree since there may be exponentially many derivations for each tree whose probabilities have to be summed up. In fact, the problem of computing the most probable tree in DOP is known to be NP hard (Sima'an 1996). Yet, the PCFG reduction in figure 4 can be used to *estimate* DOP's most probable parse tree by a Viterbi *n*-best search in combination with a CKY parser which computes the *n* most likely derivations and next sums up the probabilities of the derivations producing the same tree. (We can considerably improve efficiency by using *k*-best hypergraph parsing as recently proposed by Huang and Chiang 2005, but this will be left to future research).

In this paper, we estimate the most probable parse tree from the 100 most probable derivations (at least for the relatively small WSJ10). Although such a heuristic does not guarantee that the most probable parse is actually found, it is shown in Bod (2000) to perform at least as well as the estimation of the most probable parse with Monte Carlo techniques. However, in computing the 100 most probable derivations by means of Viterbi it is prohibitive to keep track of all subderivations at each edge in the chart. We therefore use a pruning technique which deletes any item with a probability less than 10^{-5} times of that of the best item from the chart.

To make our parse results comparable to those of Klein and Manning (2002, 2004, 2005), we will use exactly the same evaluation metrics for unlabeled precision (UP) and unlabeled recall (UR), defined in Klein (2005: 21-22). Klein's definitions slightly differ from the standard PARSEVAL metrics: multiplicity of brackets is ignored, brackets of span one are ignored and the bracket labels are ignored. The two metrics of UP and UR are combined by the unlabled f-score F1 which is defined as the harmonic mean of UP and UR: F1 = 2*UP*UR/(UP+UR). It should be kept in mind that these evaluation metrics were clearly inspired by the evaluation of *supervised* parsing which aims at mimicking *given* tree annotations as closely as possible. Unsupervised parsing is different in this respect and it is questionable whether an evaluation on a pre-annotated corpus such as the WSJ is the most appropriate one. For a subtle discussion on this issue, see Clark (2001) or Klein (2005).

3 Experiments

3.1 Comparing U-DOP to previous work

Using the method described above, our parsing experiment with all p-o-s strings from the WSJ10 results in an f-score of 78.5%. We next tested U-DOP on two additional domains from Chinese and German which were also used in Klein and Manning (2002, 2004): the Chinese treebank (Xue et al. 2002) and the NEGRA corpus (Skut et al. 1997). The CTB10 is the subset of p-o-s strings from the Penn Chinese treebank containing 10 words or less after removal of punctuation (2437 strings). The NEGRA10 is the subset of p-o-s strings of the same length from the NEGRA corpus using the supplied converson into Penn treebank format (2175 strings). Table 1 shows the results of U-DOP in terms of UP, UR and F1 compared to the results of the CCM model by Klein and Manning (2002), the DMV dependency learning model by Klein and Manning (2004) together with their combined model DMV+CCM.

Model	English (WSJ10)			German (NEGRA10)			Chinese (CTB10)		
	UP	UR	F1	UP	UR	F1	UP	UR	F1
CCM	64.2	81.6	71.9	48.1	85.5	61.6	34.6	64.3	45.0
DMV	46.6	59.2	52.1	38.4	69.5	49.5	35.9	**66.7**	**46.7**
DMV+CCM	69.3	88.0	77.6	49.6	89.7	63.9	33.3	62.0	43.3
U-DOP	**70.8**	**88.2**	**78.5**	**51.2**	**90.5**	**65.4**	**36.3**	64.9	46.6

Table 1. Results of U-DOP compared to previous models on the same data

Table 1 indicates that our model scores slightly better than Klein and Manning's combined DMV+CCM model, although the differences are small (note that for Chinese the single DMV model scores better than the combined model and slightly better than U-DOP). But where Klein and Manning's combined model is based on both a constituency and a dependency model, U-DOP is, like CCM, only based on a notion of constituency. Compared to CCM alone, the all-subtrees approach employed by U-DOP shows a clear improvement (except perhaps for Chinese). It thus seems to pay off to use all subtrees rather than just all (contiguous) substrings in bootstrapping