3.2 Verb Groups

To transform verb groups from PS to MS, the transformation algorithm, $\tau_v(\Delta)$, starts by identifying all auxiliary verbs in a sentence. These will belong to the set A and are processed from left to right. A word $w_{aux} \in A$ iff $w_{main} \xrightarrow{AuxV} w_{aux}$, where w_{main} is the main verb. The transformation into MS reverses the relation between the verbs, i.e., $w_{aux} \xrightarrow{AuxV} w_{main}$, and the former head of w_{main} becomes the new head of w_{aux}. The main verb can be located on either side of the auxiliary verb and can have other dependents (whereas auxiliary verbs never have dependents), which means that dependency relations to other dependents of w_{main} may become non-projective through the transformation. To avoid this, all dependents to the left of the rightmost verb will depend on the leftmost verb, whereas the others will depend on the rightmost verb.

Performing the inverse transformation for verb groups, $\tau_v^{-1}(\Delta)$, is quite simple and essentially the same procedure inverted. Each sentence is traversed from right to left looking for arcs of the type $w_{aux} \xrightarrow{AuxV} w_{main}$. For every such arc, the head of w_{aux} will be the new head of w_{main}, and w_{main} the new head of w_{aux}. Furthermore, since w_{aux} does not have dependents in PS, all dependents of w_{aux} in MS will become dependents of w_{main} in PS.

4 Experiments

All experiments are based on PDT 1.0, which is divided into three data sets, a training set (Δ_t), a development test set (Δ_d), and an evaluation test set (Δ_e). Table 1 shows the size of each data set, as well as the relative frequency of the specific constructions that are in focus here. Only 1.3% of all words in the training data are identified as auxiliary verbs (A), whereas coordination (S and C) is more common in PDT. This implies that coordination transformations are more likely to have a greater impact on overall accuracy compared to the verb group transformations.

In the parsing experiments reported in sections 4.1–4.2, we use Δ_t for training, Δ_d for tuning, and Δ_e for the final evaluation. The part-of-speech tagging used (both in training and testing) is the HMM tagging distributed with the treebank, with a tagging accuracy of 94.1%, and with the tagset compressed to 61 tags as in Collins et al. (1999).

Data	#S	#W	%S	%C	%A
Δ_t	73088	1256k	3.9	7.7	1.3
Δ_d	7319	126k	4.0	7.8	1.4
Δ_e	7507	126k	3.8	7.3	1.4

Table 1: PDT data sets; S = sentence, W = word; S = separator, C = conjunct, A = auxiliary verb

T	AS
τ_c	97.8
τ_{c^*}	98.6
τ_{c+}	99.6
τ_{c+^*}	99.4
τ_v	100.0

Table 2: Transformations; T = transformation; AS = attachment score (unlabeled) of $\tau^{-1}(\tau(\Delta_t))$ compared to Δ_t

MaltParser is used with the parsing algorithm of Nivre (2003) together with the feature model used for parsing Czech by Nivre and Nilsson (2005). In section 4.2 we use MBL, again with the same settings as Nivre and Nilsson (2005),[3] and in section 4.2 we use SVM with a polynomial kernel of degree 2.[4] The metrics for evaluation are the attachment score (AS) (labeled and unlabeled), i.e., the proportion of words that are assigned the correct head, and the exact match (EM) score (labeled and unlabeled), i.e., the proportion of sentences that are assigned a completely correct analysis. All tokens, including punctuation, are included in the evaluation scores. Statistical significance is assessed using McNemar's test.

4.1 Experiment 1: Transformations

The algorithms are fairly simple. In addition, there will always be a small proportion of syntactic constructions that do not follow the expected pattern. Hence, the transformation and inverse transformation will inevitably result in some distortion. In order to estimate the expected reduction in parsing accuracy due to this distortion, we first consider a pure treebank transformation experiment, where we compare $\tau^{-1}(\tau(\Delta_t))$ to Δ_t, for all the different transformations τ defined in the previous section. The results are shown in table 2.

We see that, even though coordination is more frequent, verb groups are easier to handle.[5] The

[3] TIMBL parameters: -k5 -mM -L3 -w0 -dID.

[4] LIBSVM parameters: -s0 -t1 -d2 -g0.12 -r0 -c1 -e0.1.

[5] The result is rounded to 100.0% but the transformed tree-