Data	Sent.	FS	Clauses
PDT dtest	604	88.24	88.23
PDT etest	704	87.98	88.64
CAC 2.0	3,669	85.68	85.76

Table 4: Parsing evaluation on the `0B1` sentences.

whole datasets, compare the *FS* column in Table 4 and the *UAS* column in Table 1.

Given this observation, we proposed the following strategy for parsing subordinated clauses and we updated the *CCP* method as follows:

1. Find the longest sequence of neighboring subordinated clauses C_1, C_2, ..., C_n so that $layer(C_{i+1}) = layer(C_i) + 1$ where $layer$ stands for a layer of embedding in a clause chart.

2. Create a sequence $S = C_1\ B_{1,2}\ C_2\ B_{2,3} \ldots C_n$ where $B_{i,i+1}$ is a boundary between C_i and C_{i+1}.

3. Using MST parse sequence S to get a dependency tree T_S.

Using the CCP strategy for parsing the `0B0` and `0B1` sentences, we can parse the `0B1B0` sentences so that we apply the CCP strategy for subordinated clauses first and subsequently for coordinated clauses. Table 5 presents the comparison of full-scale parsing and CCP.

Data	Sent.	FS	CCP
PDT dtest	166	81.72	82.98
PDT etest	160	81.98	84.22
CAC 2.0	885	79.68	80.84

Table 5: Parsing evaluation on the `0B1B0` sentences.

5.3 CCP as Full-scale Parsing

We have learned from the experiments that

1. it is efficient to parse coordinated clauses individually and connect their trees subsequently;

2. it is effective to parse a sequence of governing and dependent clauses at once.

Therefore we proposed and evaluated a final algorithm for dependency parsing that exploits sentence clause charts and a given dependency parser. The algorithm works in iterations. In each iteration, at least one layer of embedding in the clause chart is eliminated using the CCP strategy for `0B0` and `0B1` clauses.

Table 6 and Table 7 present the final comparison of full-scale parsing and the CCP strategy. The figures in Table 6 exclude simple sentences (one-clause sentences) from evaluation. We achieved an average 0.97% improvement in UAS when parsing all the sentences in the treebanks.

Data	Sent.	FS	CCP
PDT dtest	2,044	83.93	84.72
PDT etest	2,339	83.84	84.64
CAC 2.0	12,756	81.99	83.42

Table 6: Parsing evaluation on the sentences containing at least two clauses.

Data	Sent.	FS	CCP
PDT dtest	4,042	84.50	85.03
PDT etest	4,672	84.32	84.87
CAC 2.0	24,709	82.68	83.64

Table 7: Final comparison of full-scale parsing and CCP.

6 Future Work

Our motivation to address the task of parsing of long sentences arises from a project of extracting entities and relations from legal texts. We plan to apply the CCP strategy on the Czech Legal Text Treebank that contains significantly large number of long sentences than PDT 3.0. Consequently, we will do an eccentric evaluation of the RExtractor system to see whether better parsing results influence the extraction.

A sentence clause structure used in our experiments was generated automatically. However, the used procedure requires gold standard dependency trees on the input. We plan to develop an automatic procedure for obtaining the clause charts. This procedure will not require gold standard dependency trees on the input. Some experiments have been already done by Krůza and Kuboň (2009). In addition, we see several differ-