When a statistically oriented parser was more efficient than a linguist: A case of treebank conversion

Kiril Ribarov, Alevtina Bémová and Barbora Hladká

Abstract

The aim of the present paper is to investigate a possibility to enlarge the data in the Prague Dependency Treebank by the data included in the Czech Academic Corpus. The Prague Dependency Treebank annotation is based on a complex three-layer scenario capturing the morphemic and syntactic properties (both of the surface and of the underlying, tectogrammatical structures) of Czech sentences. The characteristics included in the Czech Academic Corpus reflect basic (mostly intra-clausal) relations between sentence elements. The integration of the Czech Academic Corpus material into the Prague Dependency Treebank implies, of course, the necessity to make the two sets of annotated data compatible. This has already been done as for the morphemic layer. The question the paper poses and attempts to answer is whether an automatic transition of the syntactic Czech Academic Corpus data into the Prague Dependency Treebank format is more effective than a direct annotation of the same texts by a statistical parser.

1 Introduction

The Czech Academic Corpus (CAC) was created manually in the 1970s and 1980s at the Institute of Czech Language under the supervision of Marie Těšitelová. The aim was to build a total of 550 thousand word tokens corpus with morphological and syntactic information in order to obtain a quantitative characteristics of contemporary Czech (Hladká and Králík, 2006). This initiative resulted in an annotated corpus with a two-layer structure:

- morphological layer,
- dependency layer, with 2 sub-layers: surface syntactic relations within a single clause, and between clauses in a complex sentence.

We need to mention that apart from the structure of a layer-to-layer correspondence with the Prague Dependency Treebank, the annotation of the corresponding layers of CAC is far from being trivial to be converted into the PDT format both in the case of the morphological layer and even more so for the case of the surface dependency one.

The morphological conversion of CAC into the PDT format came prior to the surface syntactic one and could not be performed immediately - a set of steps was needed to anticipate the morphological annotation of PDT. One set of these steps was a reconstruction of missing sentence identification, missing digit tokens (currently a predefined symbol stands at this position) and missing punctuation (inserted manually). Secondly, the format of the data needed to be adjusted to the PDT one. More information on the morphological conversion and the description of the first version of the newly converted CAC[1] (see below) can be obtained from (Hladká *et al*, 2006).

Since the morphological conversion has already been completed, we may assume that the morphological layer of CAC is compliant to the morphological layer of PDT. Such a preprocessing of the sentences is important with respect to the syntactic annotation that is to follow. Furthermore, this will allow for a full usage of the PDT technologies in the conversion of the dependency structures of CAC to PDT-like analytical trees (Hajič *et al.*, 1999). In

[1] http://ufal.mff.cuni.cz/rest

addition, we had a possibility to cooperate with some of the PDT annotators. Another advantage is the existence of various analytical parsers for Czech, trained on PDT. All these resources have been analyzed in order to select the best strategy for the syntactic conversion of CAC to PDT.

The experience collected during this work, we believe, can be used whenever one faces the following question: Given an existing treebank, which is the most efficient way to expand it? In our case the set of sentences by which we want to expand the existing treebank is previously annotated with a different annotation scheme, but following the same theoretical linguistic background.

2 Basic facts about the Czech Academic Corpus

2.1 The size and the structure of CAC

Historical background of CAC and notes advertising the release of CAC version 1.0 are given in (Hladká, this volume). Here we provide basic characteristics of CAC. CAC consists of 180 texts (documents) which belong to three different categories and are either written or spoken (transcription):

- journalistic style,
- scientific style,
- administrative style.

Table 1 includes recapitulative characteristics of CAC 1.0

STYLE	FORM	#DOCUMENTS	#SENTENCES	#WORD TOKENS
journalistic	written	52	10,234	189,435
journalistic	spoken	8	1,433	28,737
scientific	written	68	11,113	245,175
scientific	spoken	32	4,576	115,853
administrative	written	16	3,362	58,697
administrative	spoken	4	989	14,235
Total	written	136	24,709	493,307
Total	spoken	44	6,998	158,825
Total	written and spoken	180	31,707	652,132

Table 1 Quantitative characteristics of CAC 1.0

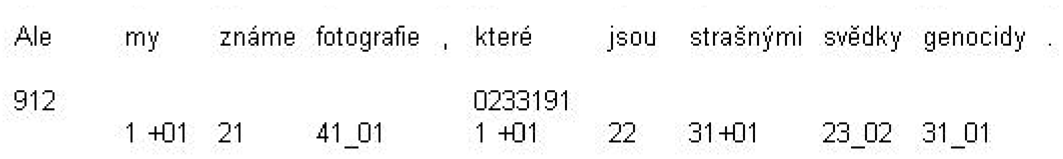
2.2 The syntactic information in CAC

The syntactic information of CAC is captured in the shape of two types of positional tags:

(i) a 6-position tag assigned to every autosemantic word of a single clause representing the dependency intra-clausal relations,

(ii) a 9-position tag assigned to the first item of each clause in a complex sentence representing the status of the given clause within the given (complex) sentence.

A detailed description of the tags is given in Appendix A (Tables A.1 and A.2). An illustrative example follows: Figure 1 presents an example of a sentence from CAC *Ale my známe fotografie, které jsou strašnými svědky genocidy*. together with the dependency tags. The word-by-word English translation of the sentence is *But we know photographs, which are terrible witnesses of_genocide*.



```
Ale   my     známe  fotografie  ,  které     jsou  strašnými  svědky  genocidy  .

912                                0233191
      1 +01  21     41_01          1 +01     22    31+01      23_02   31_01
```

Figure 1 The Czech sentence *Ale my známe fotografie, které jsou strašnými svědky genocidy* annotated by the CAC positional syntactic tags

One may notice the added punctuation tokens (comma, full stop) and as well one may notice the word *Ale* which has no 6-position dependency tag in CAC. The “`912      `″ and “`0233191  `″ stand for the 9-position tags (i.e. tags for the first word in the clause), while the others stand for the 6-position ones (for intra-clausal relations). In this simple case the syntactic structure can be obtained in a rather straightforward way as based on the interpretation of the given tags.

A closer look at the 9-position tags reveals the following:

- the tag “`912      `″: `91` should be read as `01` (`9` stands as an indication of the first clause of the sentence) and its type is a main clause (3rd position value = `2`).
- the tag “`0233191  `″: `02` means that this is the beginning of the second clause, 3rd position = `3` indicates that the clause is relative, 4th position = `3` further specifies the relative clause as attributive, 5th position = `1` indicates that the word is dependent on the token which is one position (e.g. immediately) to the left and is a noun (i.e. on *fotografie*; note that the comma does not count since this node was originally not present in CAC), and the last two positions `91` indicate the number of the governing clause which in this case is *Ale my známe fotografie*.

Table 2 presents commented examples of the 6-position tags from Figure 1:

TAG	1ST POSITION	2ND POSITION	3RD, 4TH AND 5TH POSITIONS	6TH POSITION
`1 +01`	1	empty	+01	empty
	Subject		1 word to the right	
`21`	2	1		
	Predicate	verbal		
`41_01`	4	1	_01	
	?	Object	1 word to the left	
`22`	2	2		
	Predicate	conjunction		
`31+01`	3	1	+01	
	?	object	1 word to the right	
`23_02`	2	3	_02	
	Predicate	nom.part.conj.	2 words to the left	

Table 2 Commented examples of the 6-position CAC tags

3 Basic facts about the Prague Dependency Treebank

3.1 The size and the overall scenario of PDT

The Prague Dependency Treebank (Hajič *et al.*, 2006a, Hajič *et al.*, 2006b) is a project for manual annotation of a substantial amount of Czech-language data with linguistically rich information ranging from morphemics through syntax and semantics/pragmatics.

The annotation in PDT 2.0[2] covers a large amount of Czech texts with interlinked morphological (so-called m-layer, 2 million words), syntactic (a-layer, 1.5 MW) and complex underlying syntactic and semantic annotation (t-layer, 0.8 MW). The data in PDT are annotated articles (non-abbreviated) from the newspapers and journals – see Figure 2.

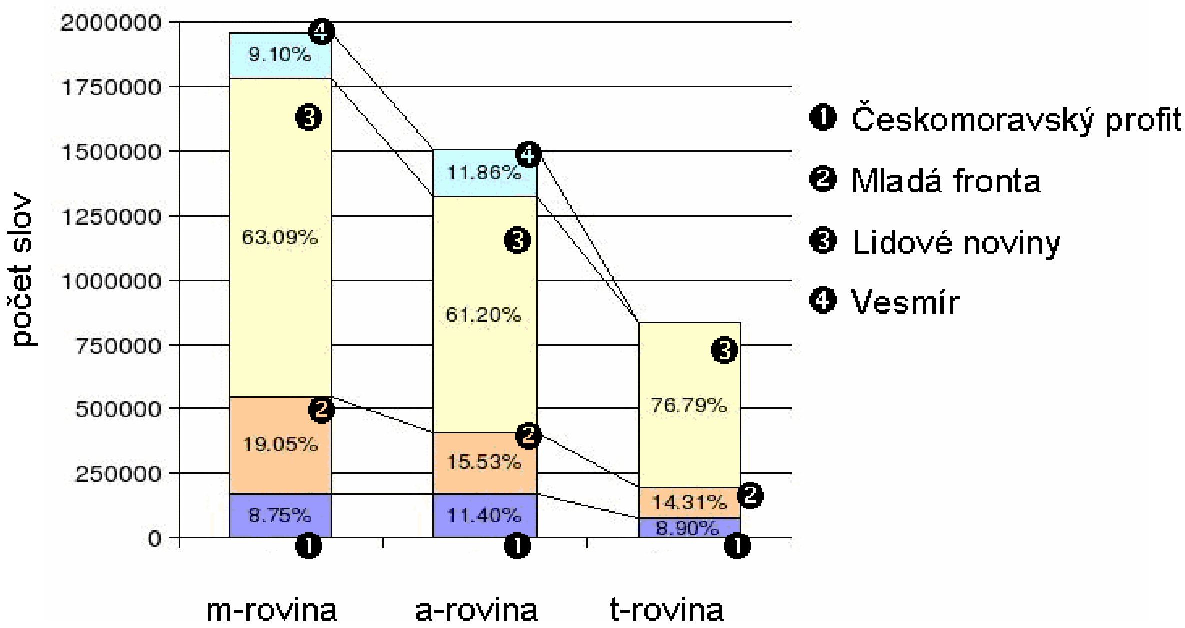


Figure 2 PDT 2.0: number of tokens from the particular sources

3.2 The PDT analytical functions

For the purpose of the integration of CAC into PDT we have chosen as the target layer of the transition the analytical layer of PDT, because quite promising procedures have already been formulated.

In order to provide the reader with more detailed information on the set of the PDT analytical functions, we give in Appendix B a list of these functions with a brief characteristic of each of them; the shape of the tree structure on the analytical layer is exemplified in Figure 3.

[2] http://ufal.mff.cuni.cz/pdt2.0

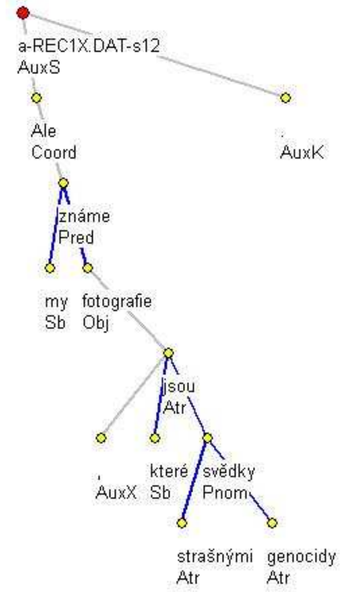


Figure 3 The analytical tree structure of the sentence *Ale my známe fotografie, které jsou strašnými svědky genocidy.*

3.3 Correspondence between CAC and PDT

The PDT analytic-layer structure of the sentence from Figure 3 can be compared with the structure obtained for the same sentence with the help of the CAC 6-position tags (see Figure 1); this comparison is illustrated in Figure 4

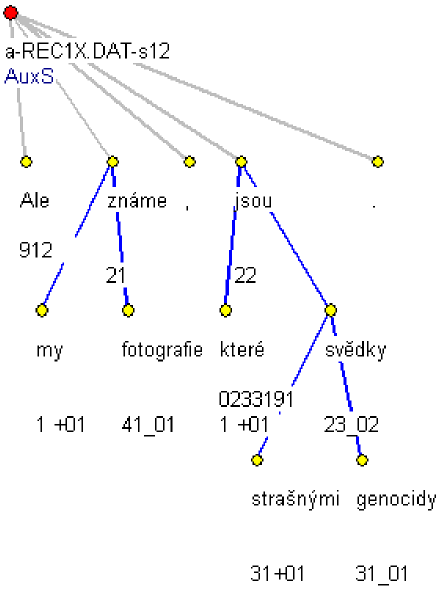


Figure 4 An “integrated” CAC and analytical PDT annotation of the sentence *Ale my známe fotografie, které jsou strašnými svědky genocidy*.

In Figure 4 there are two syntactically unattached nodes (*Ale* and comma; the final punctuation is always attached to the technical root) and the two subtrees (with their own roots attached to the technical one) are not connected into a single tree, i.e. the true root of the analytical tree has not been determined.

The information on the 9-postion tag cannot be used directly and the usage of the 9-position tag depends on the syntactic relations it codes as well as on the representation of the syntactic information in the analytical tree. In this example it is not *které*, but its governor *jsou* which is connected to *fotografie*. In the final stage the comma is attached to *jsou* and *Ale* becomes the root as visible in Figure 3.

This simple example of the correspondence between the dependency tags in CAC and the analytical functions in PDT illustrates how the correspondences of the subject, the predicate and the object can be established. One should keep in mind that CAC marks only the dependency relations within a single clause.

Figure 5 to Figure 8 display examples of a more and more complex nature than the previous one. The thick edges are those which result from the 6-position tags. It can be observed that these dependencies are mostly present after the conversion into the PDT analytical structure (the tree to the right). The English counterparts here and in the sequel are literal translations.

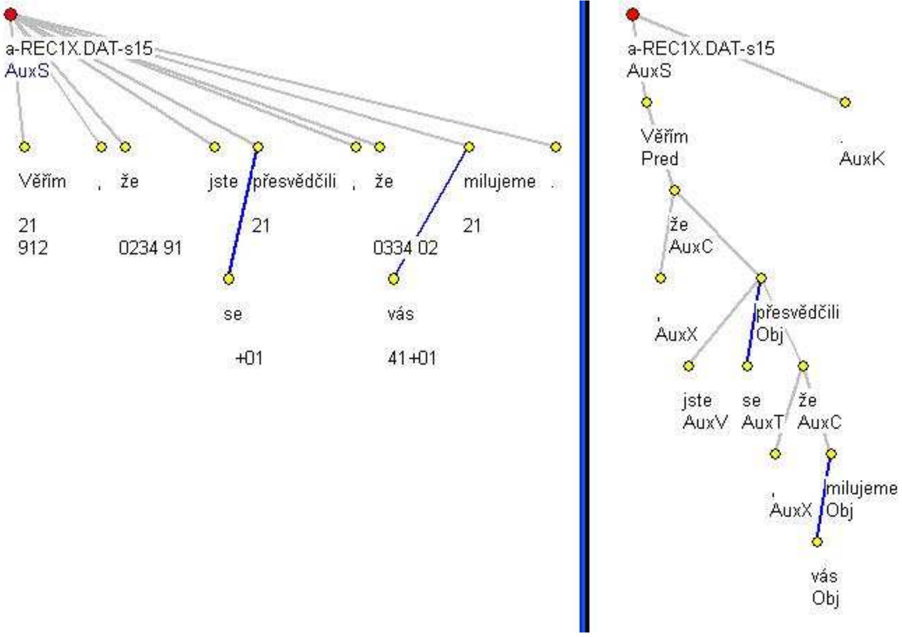


Figure 5 [*Cz*] Veřím , že jste se přesvědčili , že vás milujeme .
[*En*] I_believe , that you *Refl* convinced , that you$_{Plural}$ we_love .

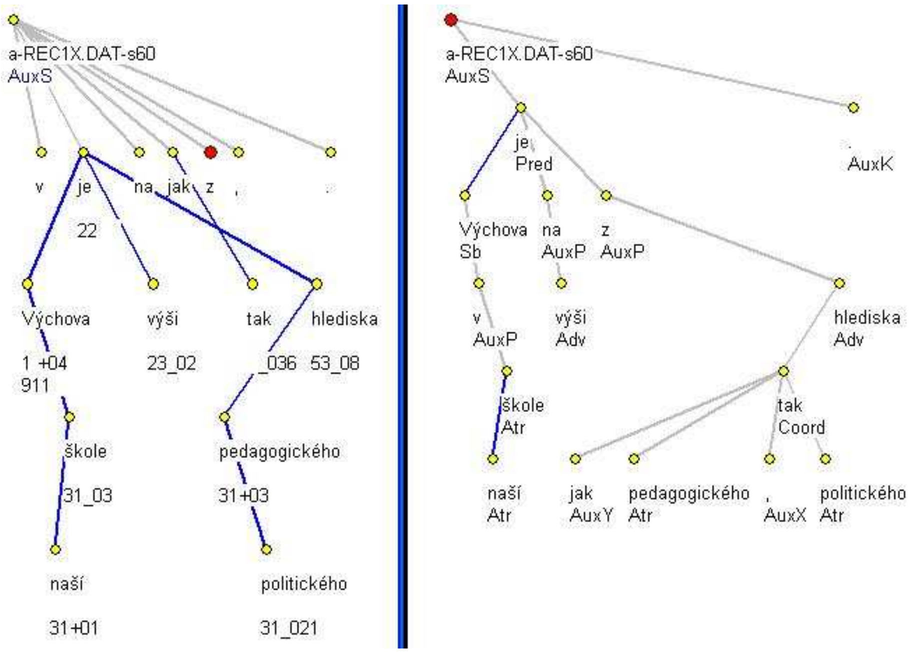


Figure 6 [*Cz*] Výchova v naší škole je na výši jak z pedagogického , tak z politického hlediska .
[*En*] Education in our school is at high_level both from the_pedagogical , as_well_as from the_political aspect .

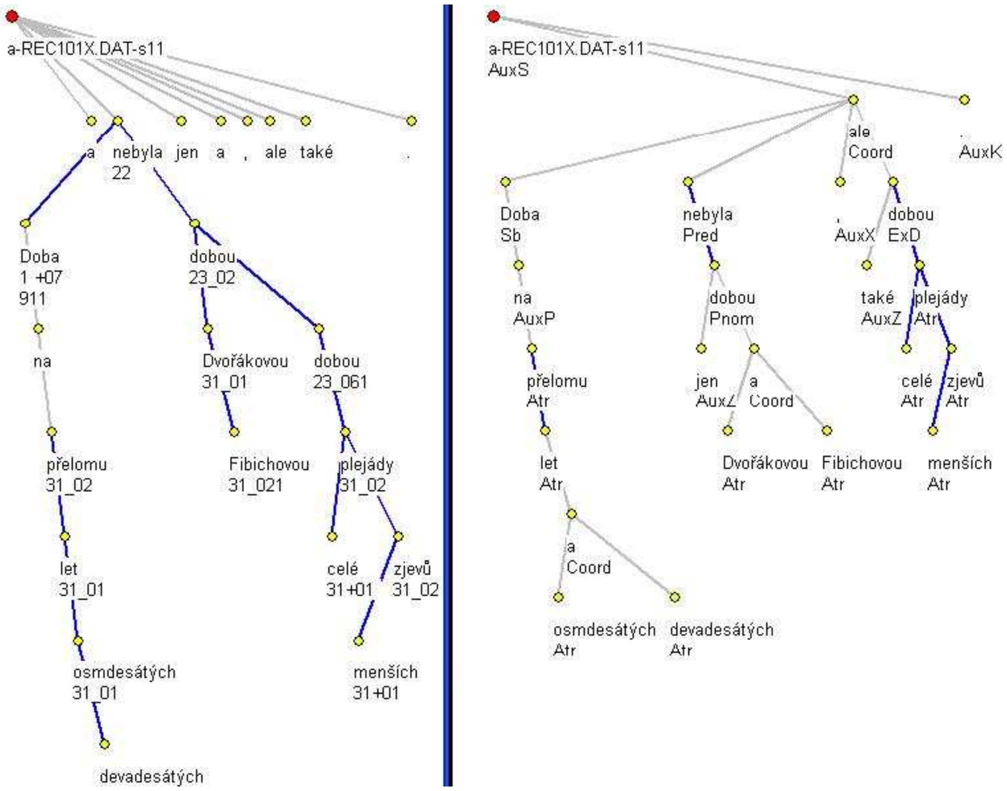


Figure 7 [*Cz*] Doba na přelomu let osmdesátých a devadesátých nebyla jen dobou Dvořákovou a Fibichovou , ale take dobou celé plejády menších zjevů .
[*En*] The-time at the_turn of_years eighties and nineties was_not only the_time of_Dvořák and of_Fibich , but also the_time of_a_whole range of_smaller personalities .

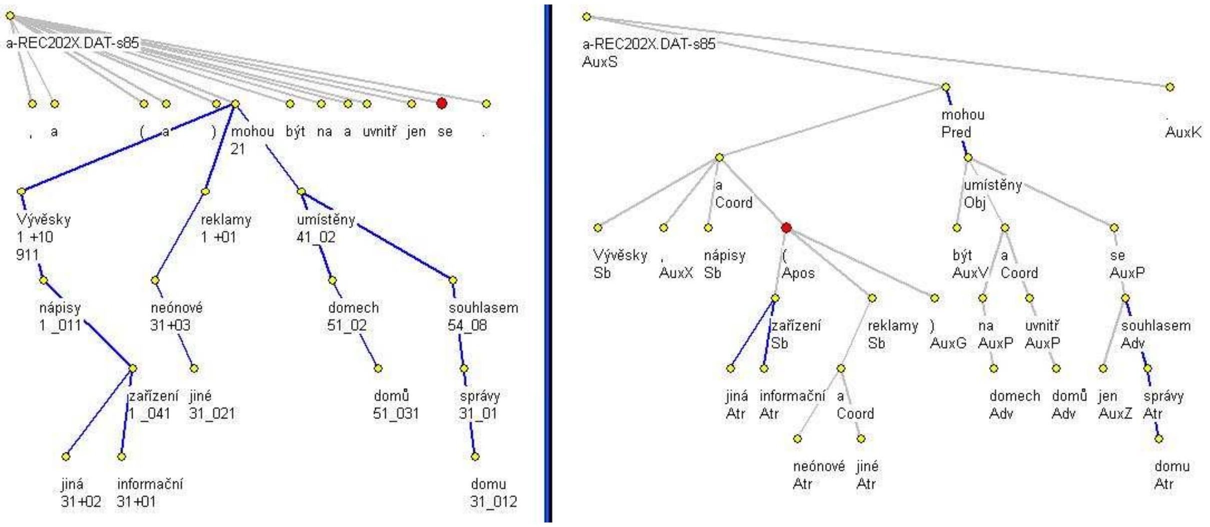


Figure 8 [*Cz*] Vývěsky , nápisy a jiná informační zařízení (neonové a jiné reklamy) mohou být umístěny na domech a uvnitř domů jen se souhlasem správy domu .
[*En*] Posters , notices and other informative devices (neon and other advertisements) can be placed on buildings and inside buildings only with approval of_the_administration of_the_building .

Besides the relatively close correspondences of the basic syntactic relations within a sentence of the type subject, predicate, object and their complements, there are significant differences in the formal representation of other analytical relations, which can be summarized as follows:

- differences in the treatment of prepositions (CAC omits them in the dependency structure),
- significant differences in the treatment of coordination and apposition,
- some elements of the sentence are not included in the dependency structure (see the left part of Figures 5 to 8).

Based on the dependency tags in CAC, a partial dependency tree of the analytical type of PDT was created automatically (programmed as a macro within the TrEd environment).[3] As stated earlier, the automatic procedure has determined correctly, in a significant majority of the cases, the predicate, the subject and the nominal groups. However, there is a number of tokens which still need to be inserted in the tree structure; these items are of the following types:

- prepositions,
- punctuation marks,
- conjunctions (coordinating and subordinating),
- reflexive particles,
- auxiliary verbs,
- certain type of adverbials,
- digit tokens.

If a sentence does not include such tokens, the resulting tree is automatically and correctly transformed into a PDT as an analytical tree. At the first glance one may be encouraged by such an observation, but these sentences occur only in 9% of the cases. The following sentence is an example of such a case. Since similar examples have been shown earlier in the text, we do not present the tree of this sentence.

[3] http://ufal.mff.cuni.cz/~pajas/tred

[Cz]	*Zájem*	*plynulosti*	*provozu*	*vyžaduje*	*stanovení*
	povinnosti	*neomezovat*	*provoz*	*bezdůvodně*	*pomalou*
	jízdou.				
[En]	*The_interest*	*of-continuous*	*traffic*	*requires*	*placement_of*
	a_requirement	*not_to_limit*	*traffic*	*without_reason*	*by_slow*
	drive.				

4 CAC to PDT conversion

According to their treatment in PDT, the punctuation marks and conjunctions, frequently present in Czech sentences, take important ‘governing’ positions in the analytical dependency tree.

With respect to the coordination on the analytical layer of PDT the following conventions apply:

- Members of the coordination are dependent on the bearers of the coordination, a comma or a conjunction.
- In the case of a coordination of more than two items, the coordinated members are dependent on the last comma or the last conjunction of the coordinated members.

In a similar vein, with respect to relations marking dependence of clauses, PDT has the following principles:

- The relative clause is dependent on the main clause through the predicate of the relative clause; this predicate hangs on the word it expands. The relative pronoun or the relative adverbial depends on the predicate of the dependent clause.
- Subordinated clause depends on its governing clause through its conjunction, while its predicate is technically dependent on the conjunction.

4.1 The conventions concerning prepositions

Apart from the annotation problems of coordination, clause dependence or marking ellipses, the placement of the preposition, given the existing dependency structure, seemed relatively straightforward: place the preposition, according to the PDT specifications, as the governing node of the sub-tree representing the prepositional phrase. Therefore the original macro converting the CAC tags into a tree structure can be enriched by a rule for the addition of the preposition. Figure 9 displays such an example with the preposition attached correctly.

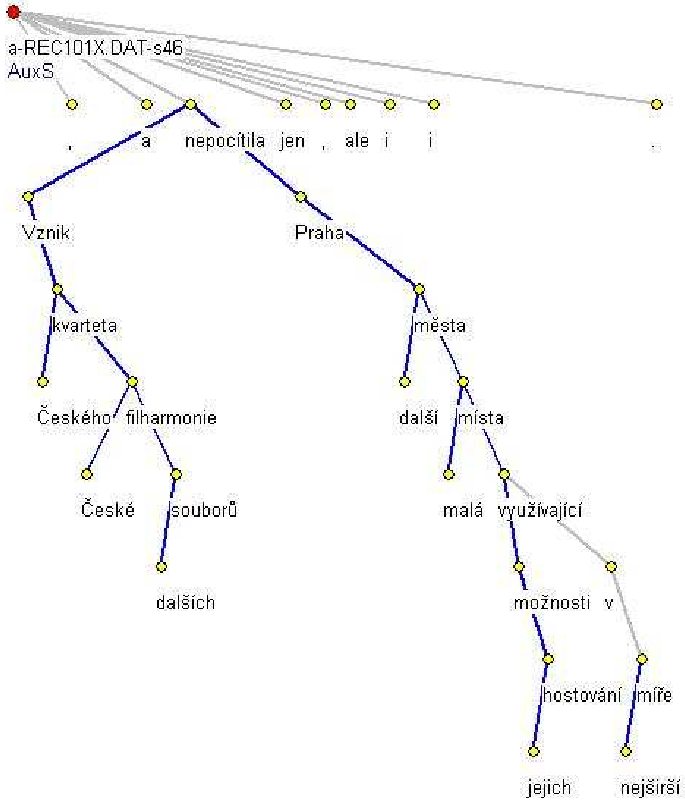


Figure 9 An example illustrating the case of preposition insertion

Figure 9 represents a partial analytical tree of the sentence:

[Cz] Vznik Českého kvarteta, České filharmonie a
dalších souborů nepocítila jen Praha, ale i další města
i malá místa využívající možnosti jejich hostování
v nejširší míře.

[En] The_origin of_the_Czech quartet, the_Czech philharmonic_orcestra and
further ensembles was_not_felt only_byPrague but also by_other cities
and small towns using the_advantages of_their visit
in the_broadest sense.

The preposition *v* (*[En] in*) was assigned by an automatic post processing after a partial analytical tree was obtained based on the CAC dependency tags. The above-stated rule for the placement of prepositions needs to be made more precise since it is not always possible to locate the root of the subtree of the prepositional phrase. The proposed steps are the following:

1. Take the token immediately to the right (in the surface representation of the sentence) to the preposition.
2. Identify this token in the dependency tree of CAC (partial trees as presented on the left-hand part of the Figures 5 to 8).
3. Traverse the path towards the root of the tree until a noun[4] (or a number) is found.

[4] Including a pronoun in this case resulted in a higher error rate, thus the cases in which the preposition governs a pronoun were handled manually. The reason for this is that frequently there is a pronoun between the preposition and a noun it belongs to and based on the given information it is not possible to distinguish whether the preposition belongs to the pronoun or to the noun. For a better understanding of this difficulty see the morphological annotation of pronouns in Hladká et al. (2006).

4. Insert the preposition into the tree between the identified noun (number) node and its governor.
5. If a suitable noun (number) node is not found the preposition remains attached to the technical root and then it is treated manually.

Except for specific cases (e.g. double prepositions, preposition attached to pronouns), this procedure placed the preposition at a correct position in the tree. The correct placement was observed in 89% of all cases involving prepositions.

In the sequel, by CAC dependencies we refer to a partial tree as in Figure 9 obtained automatically from the CAC dependency tags and automatically post processed prepositions. To avoid misunderstanding we use dependencies to relate to the syntactic information present in CAC, while analytical tree and analytical relations will be used for objects from the analytical layer of PDT. If not stated otherwise, by PDT we refer to the PDT version 2.0.

4.2 Manual vs. automatic parsing

In a similar vein to the prepositional rule, a rule for automatic post-placement of auxiliary verbs within analytical verb forms can be developed. Nevertheless, this is not the case for other syntactically unassigned tokens, such as the placement of reflexive particles (although this case may seem non to be difficult, it is enough to have a look at the treebank and one will immediately notice the obstacles) or even worse, building of coordinations.

Experiments were performed in order to decide on an effective annotation strategy. The following alternatives were taken into consideration:

1. manual assignment of missing nodes, given the CAC dependencies,
2. application of automatic analytical parsing, trained on PDT, with a post-manual correction,
3. creation of linguistically motivated rules for specific problems (reflexives, coordination, improvement of the preposition assignment, assigning adverbials) with a post-manual correction.

Although the last alternative may seem at the first glance a reasonable solution, the time to annotate, as well as the necessary manual check up afterwards, and the time needed to write the rules and test them does not speak in favor of this alternative. Before this alternative was discarded, we have determined to spend two months on tests with hand written linguistically based rules. No significant improvement on the accuracy of the analytical trees was obtained. Instead, the analysis of the CAC data projected a long lasting process, which would need a thorough manual check up. Therefore, our three alternatives were reduced to the first two.

The manual building of the dependency tree consists of two consecutive steps, i.e. creation of the dependency tree structure and the assignment of analytical functions to all nodes of the tree. In case of an automated parsing, both structure and tags are assigned simultaneously. The automatic parser used in our experiments is the Maximum Spanning Tree dependency parser, the currently best parser for Czech (McDonald *et al*, 2005) with the accuracy rate on unlabeled dependency structures of 84.6%.

The first set of experiments was performed on CAC dependencies (as described above, alternative 1), while the second one was performed on automatically generated trees with the MST parser (alternative 2). For the second alternative, the parser takes a morphologically annotated input sentence (from CAC 1.0) and does not pay attention to the dependency tags in CAC. In experimenting with both alternatives, sentences were selected from the three different types of texts present in CAC. For alternative 2, the parser was trained on the training set of PDT.

The evaluation of the alternative 1 is summarized in Table 3 and Table 4.

FILE NAME	# SENTENCES	#TOKENS/AVG.SNT.LENGTH	TEXT TYPE	TIME FOR ANNOTATION
a02w	159	3074 / 19	legal	6 hours
n01w	175	3130 / 18	newspaper	7 1/6 hours
s01w	141	3110 / 22	scientific	5 1/3 hours
Total	475	9314 / -		18 ½ hours

Table 3 Experience on annotation of the a02w, n01w, s01w documents

FILE NAME	# SENTENCES	#TOKENS/AVG.SNT.LENGTH	TEXT TYPE	TIME FOR ANNOTATION
a03w	123	3094 / 25	legal	6 ½ hours
n02w	209	3160 / 15	newspaper	6 hours
s02w	187	3142 / 17	scientific	6 ½ hours
Total	519	9396 / -		19 hours

Table 4 Experience on annotation of the a03, n02, s02w documents

These statistics cannot be explained directly without paying a careful attention to the number of sentences and their length and to the text types and their syntactic characteristics. The following comments characterize the text types:

- a02w: Text on the regulations concerning the use of apartments; relatively short sentences with a simple syntactic structure.
- n02w: Newspaper texts on different topics, ranging from political comments to sport news; sentences with a simple syntactic structure.
- s01w: Texts about Czech music with a clear and logical way of narration.
- a03w: Legal text on employment regulations; the text consists of syntactically very difficult and rich constructions with a large number of relative and subordinated clauses and many ellipses. A significant portion of the sentences consists of very long sentences (frequently 50 and more tokens) and the distance between dependent tokens is often large as well (it is not an exception that the distance exceeds 15 tokens).
- n02w: Newspaper texts on agricultural and quality evaluations; the sentences are not very complicated but contain ellipses.
- s02w: Scientific texts about human behavior; sentences with a rich syntactic structure, but of an understandable nature and with a large amount of complex structured coordination. This file contains also incomplete sentences.

It is clear that the sentences from Table 4 are syntactically more difficult than the sentences in Table 3[5]. The files were selected without a prior analysis of their contents. Longer sentences require more time for a manual check up also due to technical limitations of the size of the tree on the computer screen. But the real difficulty that is directly projected in the time is when the sentence has a complicated logical structure which happens when the ellipses or nested coordinations are present in sentences. It is also often difficult for the human annotator to determine the correct analytical structure; in such cases other annotators were consulted.

A trained linguist and annotator, who was a part of the annotation team of PDT, did the analysis manually. Despite the differences in the text sets of the alternatives, we may conclude that:

- The time in alternative 1 has no advantage over the time of alternative 2.

[5] There was no special intention to split the test files into two groups.

- The time spent for manual post correction of the trees obtained from CAC is of the 'same amount' as the time spent on manual correction of trees obtained by the MST parser without paying attention to the syntactic information in CAC.
- The MST parser returns more frequently a completely correct sentence (in 35% of the cases which is to be compared to the 9% of sentences converted correctly and directly from CAC using its dependency tags).
- Assuming that the CAC sentences are to be added to PDT, the parser can be trained on the whole PDT and not only on its training set[6] as in our experiment. It is expected that this will slightly increase the success rate of the parser. As portions of new CAC sentences are manually post-processed and annotated, the parser can be retrained on a training set containing also these sentences and thus its output can be tuned better.

4.3 Analytical functions

At this stage the sentences have their analytical tree structure, but are not labeled with analytical functions. The analytical functions, during a manual annotation as well as during an automatic one, are assigned after the tree structure has been built. Although our automatic parser outputs directly a labeled analytical tree, this procedure is internally processed as two steps with the tree structure determined first.

The assignment of the analytical functions can be done in the following three ways:

(i) transferring CAC tags into PDT analytical functions (for the cases where this is possible), with a manual post-correction

(ii) using a macro which operates on the analytical tree, with manual post-correction; this macro is available from the annotation of PDT and is based on automatically acquired decision trees similar to those described in (Sgall, Žabokrtský, Džeroski 2002),

(iii) applying automatic assignment of analytical functions, with manual post correction; the automatic labeling is obtained with the MST parser.

In both of the first two cases the annotator receives nodes which have no analytical functions. Therefore, the annotator needs to check the assigned analytical functions and at the same time has to add the missing ones. In the third case the annotator only checks and corrects the automatic tree labeling by the analytical functions. We have also observed that the automatic macro assigns analytical functions correctly in almost all cases where the CAC dependency tags can as well be converted directly. Hence, way 1 is a subset of way 2 and our selection is reduced to the last two alternatives.

In the case of automatic assignment of analytical functions with the MST parser the following are the sources of the most common errors:

- **Ellipses**

  Nodes with a missing governor are assigned the analytical function ExD, see the example in Figure 10. The first part of the sentence (up to the first comma) is a complete one with Obj, Sb, Pred, Atr and Adv, while in its second half there is no predicate. The subjects *ředitel* as well as *pracovník* (mutually coordinated) and also *ředitelství* have a missing governor. It is exactly in these cases where the automatic procedure makes most of the mistakes and assigns Sb, Adv, Atr, Obj instead.

[Cz] Pracovníky$_{Obj}$ *do pracovního*$_{Atr}$ *poměru*$_{Adv}$ *přijímá*$_{Pred}$
ředitel$_{Sb}$ *závodu*$_{Atr}$*, na podnikovém ředitelství*$_{Exd}$
ředitel$_{Exd}$ *podniku, další pracovník*$_{Exd}$ *písemně*
zmocněný ředitelem závodu.

[6] It is not expected that this will raise the parser accuracy by more than 1%.

[En] *Emplyees*^{Acc} *to working relation accepts*
head^{Nom} *of_institution, at firms' head_offices*^{Nom}
the_head of_the_company, other employee^{Nom} *in-writing*
authorized by_head of_institution.

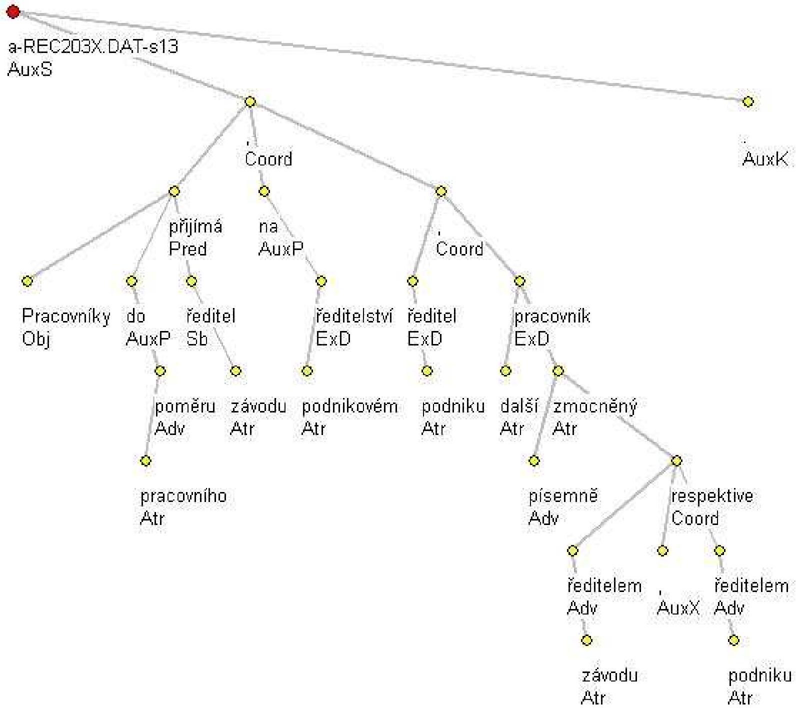


Figure 10 An analytical tree structure of the sentence *Pracovníky*$_{Obj}$ *do pracovního*$_{Atr}$ *poměru*$_{Adv}$ *přijímá*$_{Pred}$ *ředitel*$_{Sb}$ *závodu*$_{Atr}$*, na podnikovém ředitelství*$_{Exd}$ *ředitel*$_{Exd}$ *podniku, další pracovník*$_{Exd}$ *písemně zmocněný ředitelem závodu.*

- **Annotation of AuxX vs. Coord in case of coordinations**

The previous sentence can be used also to demonstrate mistakes of this type because in the automatic procedure the commas frequently receive the AuxX analytical function instead of the Coord one. In this sentence both commas should receive the Coord tag since both of them govern different types of coordination and therefore this is not a case of multiple coordination (with which Coord is assigned only to the last punctuation or last comma token).

- **Reflexive particles**

For this type of mistakes it is useful to recall that the reflexive particle "se" can typically receive either AuxT or AuxR or Obj, while the reflexive particle "si" can receive AuxT or Obj or Adv or AuxO. This demonstrates the ambiguity of the particles. There are cases when more possibilities are plausible and the annotator needs to decide which one to choose, as in "přesvědčit se" (En. "convince" *Refl*) when the particle can obtain either AuxT or AuxR or Obj.

- **False prepositions**

Although secondary prepositions are labeled correctly, mistakes occur in ambiguous cases as in[7]

[7] We list the examples with the correctly assigned analytical functions.

[Cz] necháváme problém stranou$_{Adv}$

[En] we_leave problem aside

[Cz] město bylo stranou$_{AuxP}$ *hlavních dopravních cest*

[En] town was away of_main transportation routes

[Cz] v$_{AuxP}$ *nediferencované podobě*$_{Adv}$ *amatérského provozu*

[En] in non-differentiated form of_amateur traffic

[Cz] v$_{AuxP}$ *podobě*$_{AuxP}$ *modelu*

[En] in_form of_model

[Cz] demonstrovat sílu umění v$_{AuxP}$ *nejširším světovém rámci*$_{Adv}$

[En] demonstrate power of_art in widest world frame

[Cz] v$_{AuxP}$ *rámci*$_{AuxP}$ *projektu*

[En] in frame of_project

It depends on the context how to label correctly situations such as:

[Cz] v dohodě s hygienikem

[En] in accordance with hygienist

In the case of assignment of analytical functions with macros, similar mistakes occur and most of them are due to ellipses and coordinations. That is why the analysis of the mistakes of the macro is not presented here.

On the basis of the above experiments and on error observations for the problem of analytical functions assignment we may conclude that the automatic transformation of the CAC tags has no advantage over the applied automatic MST labeling.

5 Conclusion

To answer the question on how to complete the CAC to PDT conversion in an effective way was our primary interest. The effectiveness is in our case measured mainly by time and cost limitations within this conversion project.

We would like to add here a judgement of the annotator who was mainly involved in the manual evaluation of the MST parser output and in the conversion of the CAC dependency tags into a PDT-like analytical tree. She preferred the automatic output of the MST parser to the CAC dependencies although automatic analytical tree results from a statistical procedure with irregular distribution of errors. This judgement can be described as stemming from a psychological factor. It is opposed to our initial expectation that the correct subtrees extracted from the 6-positional tags would create a more comfortable environment for the annotators as such results had been expected to be more regular and therefore more reliable in terms of types of errors.

Taking into consideration the results and measurements presented in this paper, it is our conclusion that the syntactic information present in CAC is not of significant help for its conversion to the PDT structure. Speaking more broadly, a pure dependency parser trained automatically on a suitable treebank with at least 84% accuracy rate and able to perform the labeling of the analytical trees is preferred to manually inserted but partial syntactic information with linguistically-based rules for post processing. One also should not forget that this is a one-off process, so linguistically motivated and hand crafted procedures would be difficult be reused in the future. Such circumstances justify our primarily time-cost decision strategy.

We are convinced that these results are of broader character, and that our work gives a solid basis for future similar studies and situations. To our best knowledge, the CAC to PDT conversion is the first case of such ‘revival’ of old data by a modern treebank.

Despite the fact that the automatic procedure was correct almost in the majority of the cases where the syntactic information could have been revealed directly from the CAC dependency tags, we would not like to say that the CAC tags should be completely discarded. They are useful for conversion of sentences which have no omitted tokens, they may also be used to check the output of the automatic parser on predicate attachment to nominal groups, as well as of a cross check of core dependency relations such as the subject, the object and the predicate in the main clause.

Last but not least, we believe that this study is encouraging for treebank expansions with completely new and unprocessed sentences given that there is a treebank of the size of PDT since, we have observed no significant difference in terms of efficient data enlargement whether or not the sentences to include have been previously analyzed with a different methodology.

Such conclusions would have not been possible without the existence of the Prague Dependency Treebank, and without it, the statistically oriented parser which helped more than the linguist would have not been trained.

6 References

Jan Hajič, Eva Hajičová, Jaroslava Hlaváčová, Václav Klimeš, Jiří Mírovský, Petr Pajas, Jan Štěpánek, Barbora Vidová Hladká and Zdeněk Žabokrtský. Prague Dependency Treebank 2.0 – Guide. Technical report, UK MFF ÚFAL Praha, 2006a.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, and Marie Mikulová. Prague Dependency Treebank 2.0 – CD-ROM. LDC2006T01, ISBN 1-58563-370-4, Linguistic Data Consortium, 2006b.

Jan Hajič, Jarmila Panevová, Eva Buráňová, Zdeňka Urešová, Alla Bémová. Annotations at Analytical Level - Instructions for annotators, UK MFF ÚFAL Praha, 1999.

Barbora Hladká, Jan Hajič, Jiří Hana, Jaroslava Hlaváčová, Jiří Mírovský, Jan Votrubec. *Czech Academic Corpus 1.0 Guide*, Karolinum - Charles University Press, 2006.

Barbora Hladká. The Czech Academic Corpus version 1.0 has been released. *This volume.*

Barbora Hladká and Jan Králík. Proměny Českého akademického korpusu, In *Slovo a slovesnost*, vol. 67, pp. 174-194, 2006.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, Jan Hajič. Non-Projective Dependency Parsing using Spanning Tree Algorithms. In *Proceedings of Human Langauge Technology Conference and Conference on Empirical Methods in Natural Language Processing (HTL/EMNLP).* Vancouver, BC, Canada, Oct. 6-8: Association of Computational Linguistics, 2005. s. 523-530.

Petr Sgall, Zdeněk Žabokrtský, Sašo Džeroski. A Machine Learning Approach to Automatic Functor Assignment in the Prague Dependency Treebank, In: Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002), eds. González Rodríguez, Manuel Paz Suárez Araujo, Carmen, Las Palmas de Gran Canaria, Spain, 5. 2002. s. 1513--1520.

Acknowledgement

Our thanks go primarily to our professors Jarmila Panevová, Eva Hajičová and Petr Sgall for their valuable suggestions and comments.

This work was supported by the grant numbers 1ET101120413 and 1ET100300517 of the Grant Agency of the Czech Academy of Science, Czech Republic.

Appendix A

The 6-position tag description is presented in Table A.1, while Table A.2 presents the structure of the 9-position tag. The 6-position tag indicates the dependency tree structure of a single clause on its 3rd, 4th and 5th position in terms of the relative distance to the governor. The relative distance does not include the tokens added later as a part of the conversion (punctuation, digit tokens): e.g., the sequence “A , B” is interpreted as “A B”.

POSITIONS: 1	**2**	**3**	**4-5**	**6**
1 Subject		**‘_’** governor to the left **‘+’** governor to the right	Number of positions/distance (in words) between the current word and its governor; immediate neighbors have distance 1.	**1** Coordination (is not assigned to the first member of the coordinated elements) **2** Complex naming of a determinative nature **3** Coordination within a complex naming **4** Other complex naming **5** Complex naming in coordination with another complex naming **6** Conj. and proverb. couple **7** non documented **8** non documented **9** Deleted governing expression **0** Eliminated governing expression
2 Predicate	**1** verbal			
	2 copule			
	3 nom. part of verbo-nominal pred.			
	4 nomin.			
	5 copula in subject-less sentence			
3 non documented	**1** attribute			
	2 apposition			
4 non documented	**1** object			
	2 complement			
5 Adverbial	**1** place			
	2 time			
	3 mood			
	4 cause			
	5 origin			
	6 author			
	7 result			
6 Clause core	**1** nominal			
	2 adjective			
	3 interjection			
	4 particle			
	5 vocative			
	6 adverbial			
	7 infinitive			
	8 verbal			
	9 verb. nominal			
	0 pronominal			
7 Trans. type (with general subject)				
8 Independent clause member				
9 Parenthesis				

Table A.1 The 6-position CAC tags description

POSITIONS: 1-2 **CLAUSE NUMBER**	**3** **TYPE**	**4** **CLAUSE TYPE**	**5** **POSITION OF THE GOVERNING NODE OF THE ATTRIBUTIVE CLAUSE**	**6-7** **NUMBER OF THE GOVERNING CLAUSE**	**8** **RELATIONS BETWEEN CLAUSES**	**9** **NON DOCUMENTED**
Clause number within a sentence; if 9 stands at position 1 it designates the first clause in the sentence	**1** Simple		**1** dependence on the immediately proceeding token (a noun) **2, 3, ..., 9** dependence on the 2nd, 3rd, ... 9th token (a noun) to the left of the relative clause **0** marks more than 9 **!** false relative clause Position is not registered in the cases of: - coordination, - for relative clauses of the types time, mood and cause, - in cases of forward links.		**1** Coordination **2** Parenthesis **3** Direct speech **5** Parenthesis in direct speech **6** Introductory clause **8** Parenthesis in an introductory sentence **!** Error in sentence structure **7** non documented	**1** non documented
	2 Main					
	3 Subordinated	**1** subject				
		2 predic.				
		3 attrib.				
		4 object.				
		5 local				
		6 time				
		7 mood				
		8 cause				
		9 complement				
	4 non documented					

Table A.2 The 9-position CAC tags description

Appendix B

ANALYTICAL FUNCTION	DESCRIPTION
Pred	Predicate
Sb	Subject
Obj	Object
Adv	Adverbial
Atv	Complement technically hung on a non-verbal element
AtvV	Complement hung on a verb, no 2nd gov. node
Atr	Attribute
Pnom	Nominal predicate, or nom. part of predicate with copula *be*
AuxV	Auxiliary verb *be*
Coord	Coordination node
Apos	Apposition (main node)
AuxT	Reflex. tantum
AuxR	Ref., neither Obj nor AuxT, Pass. refl.
AuxP	Primary prepos., parts of a secondary prep.
AuxC	Conjunction (subord.)
AuxO	Redundant or emotional item, ‘coreferential’ pronoun
AuxZ	Emphasizing word
AuxX	Comma (not serving as a coordinating conj.)
AuxG	Other graphic symbols, not terminal
AuxY	Adverbs, particles not classed elsewhere
AuxS	Root of the tree (#), the only added node, technical node
AuxK	Terminal punctuation of a sentence
ExD	A technical value for a deleted item
AtrAtr	An attribute of any of several preceding (syntactic) nouns
AtrAdv	Structural ambiguity between adverbial and adnominal (hung on a name/noun) dependency without a semantic difference
AdvAtr	Same as AdvAtr, with reverse preference
AtrObj	Structural ambiguity between object and adnominal dependency without a semantic difference
ObjAtr	Same as AtrObj, with reverse preference

Table B.1 The PDT analytical functions description