PDT 2.0 – Guide

Jan Hajič, Eva Hajičová, Jaroslava Hlaváčová,
Václav Klimeš, Jiří Mírovský, Petr Pajas, Jan Štěpánek,
Barbora Vidová Hladká, and Zdeněk Žabokrtský

June 20, 2006

Contents

Chapter 1

Introduction

This guide introduces the Prague Dependency Treebank, version 2.0 (PDT 2.0). The guide allows you to become quickly familiar with the basic ideas as well as contents of PDT 2.0. It provides an overview of its data and tools, together with links to more extensive documentation, including tutorials, formal specifications and further references. This document exists both in an HTML and a PDF version.

The website of PDT 2.0 is <http://ufal.mff.cuni.cz/pdt2.0>. You can also view the web page <http://ufal.mff.cuni.cz/pdt2.0update>, where possible corrections of the data, improved versions of the tools etc. will be published.

1.1 What is PDT 2.0

The Prague Dependency Treebank (PDT) is an open-ended project for manual annotation of substantial amount of Czech-language data with linguistically rich information ranging from morphology through syntax and semantics/pragmatics and beyond.

PDT version 2.0 is a sequel to version 1.0; PDT version 1.0 contains manual annotation of morphology and (surface) syntax (see <http://ufal.mff.cuni.cz/pdt/> or the web page of Linguistic Data Consortium (LDC), <http://www.ldc.upenn.edu>, Catalog No. LDC2001T10). Version 2.0 adds the underlying syntax and semantics, topic/focus, coreference and lexical semantics based on a valency dictionary to the surface syntax and morphology that have been at the core of version 1.0. The corrections of version 1.0 are also included in version 2.0, even with the old data format preserved for those who have already invested into its use.

The annotation in PDT 2.0 covers a large amount of Czech texts with interlinked morphological (2 million words), syntactic (1.5 MW) and complex underlying syntactic and semantic annotation (0.8 MW). The corpus itself now uses the latest annotation technology (standoff annotation using XML, RelaxNG—see Section 3.4 and the whole Chapter 3, "Data").

PDT 2.0 is based on the long-standing Praguian linguistic tradition and adapted for the current Computational Linguistics research needs (see also Section 1.2). Software tools for corpus search, annotation and language analysis are included. Extensive documentation (in Czech and English) is provided as well.

This version of PDT concludes a 10-year period of development at the Institute of Formal and Applied Linguistics (ÚFAL) and its Center for Computational Linguistics (see Section 1.3). Recently, the project has been complemented with the publication of the Prague Arabic Dependency Treebank, <http://www.ldc.upenn.edu>, Catalog No. LDC2004T23 and a parallel Prague Czech-English Dependency Treebank, <http://www.ldc.upenn.edu>, Catalog No. LDC2004T25. The former project demonstrates that the Czech specifications can be adapted to a typologically different language and the latter one builds on the manual annotation of the Penn Treebank corpus and it is geared towards Machine Translation experiments between the two languages.

PDT 2.0 has had two purposes:

- first, to map the theoretical achievements of the Prague Linguistic School to real language data, and thus explicitly test and preserve the theory of the dependency-based *Functional Generative Description (FGD)* (see also Section 1.2) not only "on paper", but applied to a very large number of real "examples";

- second, to allow for machine learning methods to be applied to yield automatic analysis and generation tools with reasonable accuracy.

Whereas the first purpose could have been served by choosing only a few examples for each linguistic phenomenon, the second one definitely needs a large number of naturally occurring sequences of sentences. The statistics obtained from them can certainly be used also for linguistic research with a distinct advantage.

The future of PDT is not completely determined at this point. There are several future directions under consideration (funding permitting, of course): adding spoken data; adding a deeper and broader annotation especially for coreference, information structure and/or discourse; annotation of another (very different) language; manual annotation of Czech/English and other parallel texts using the same (tectogrammatical) representation; and adding another layer (contents-based knowledge representation).

1.2 Historical background of the project

Prague School of Functional and Structural Linguistics is distinguished from other European schools of linguistic structuralism—among other things—by its openness to new trends and ideas. The history of the School formally dates back to 1926, when the Prague Linguistic Circle was founded by such prominent linguists as Vilém Mathesius, Roman Jakobson, and Bohumil Trnka. The research paved the way into several directions—phonology was perhaps the first internationally highly appreciated domain. Soon there also appeared (with a positive international acceptance) original contributions to language typology, word-formation, functional stratification of language, to such general linguistic issues as that of the distinction of core and periphery in the language system and, last but not least, attempts at a systematic account of the information structure of the sentence (functional sentence perspective, topic-focus articulation).

The Prague Linguistic Circle did not restrict its activities geographically; there were several linguists abroad who openly avowed the Circle's tenets and worked in their intentions. One of them was Lucien Tesnière, a French linguist, who can be duly called "the Father of dependency syntax". Tesnière's approach had found a very positive response also outside the Circle, especially in the work of the Czech syntactician Vladimír Šmilauer, whose *Novočeská skladba* (*Syntax of Modern Czech*, 1947) is a non-omissible source of information for all those who study Czech syntax.

The Prague School inspiration has found a continuation also in the new linguistic paradigm of explicit description of language, namely in the Functional Generative Description (FGD) as proposed by Petr Sgall in the 1960s and elaborated since then by him and his collaborators (for a most complex treatment, see the book *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*, 1986). There are three important distinctive features of the FGD framework:

- inclusion of an underlying syntactic layer (tectogrammatics) into the linguistic description;
- use of dependency syntax;
- a specification of a formal account of the information structure (topic-focus articulation) of the sentence and its integration into the description.

1.3 Development of the project

The project started, in fact, in the lobby of a small hotel in Dublin, Ireland, in the end of March in 1995 during the 7th conference of the European Chapter of the ACL. A small group of us decided to pursue a project similar to the English Penn Treebank project which came out then not so long ago, but based on the Praguian dependency tradition, with full morphological analysis and with the perspective of gradual enrichment of the annotation (for more on the project context, see Section 1.2).

Funding had to be secured first; we were lucky to get two grants of the Czech Grant Agency and one Ministry of Education projects simultaneously, starting in 1996: one smaller grant to write the specification of the treebank, one multi-institutional project to support the Czech National Corpus (our source of raw texts), and finally, a project called the "Linguistic Data Lab" to get the annotation itself going.

The specification called for a three-layer annotation scenario, with morphological, analytical and tectogrammatical layers of annotation. Except for the morphological layer, which was designed to use the existing Czech tagset, the annotation guidelines were only sketchy, with the understanding that they would be developed in parallel with the annotation as new phenomena and problems would be discovered. Nevertheless, some basic principles were taken as the "unbreakable" rules:

- morphological annotation will be applied to individual tokens; no attempt will be made to analyze e.g. complex verb forms,
- the tagset used in the existing morphological dictionary for Czech, developed at ÚFAL, will be used directly for annotation,
- the unit of the surface-syntactic annotation (the analytical layer) will also be a token, with a 1:1 correspondence to the morphological layer units; no "traces", substitutes for ellipsis or anything like it would be inserted into the annotation,
- dependency-style annotation will be used not only on the underlying syntactic layer (the tectogrammatical one), but also on the analytical layer,
- the tectogrammatical annotation will include all what the theory has to offer, i.e. topic/focus, coreference, and other detailed annotation; "inserting" and "deleting" nodes (with respect to the lower layers) will be allowed to match the theory and the desired purpose of underlying representation,
- valency will be taken into account when determining verb (noun, adjective) dependent's function.

Moreover, some further decisions were made. The data markup format was designed as the extension of the proprietary SGML format called *CSTS* used in the Czech National Corpus. Then, the organization of the annotation had to be determined: we started annotation of the lower two layers simultaneously (morphology and analytical syntax); the tectogrammatical layer annotation had been postponed until the first two layers were finished. Furthermore, tools were developed for the annotation to proceed. Among them, `Graph`, the grafical tree editor has been using our proprietary annotation format (called *FS*), a non-SGML but quite general and space-saving one.

The annotation of the morphological and analytical layers was performed mainly by students with linguistic background. The lack of complete guidelines at the analytical layer required weekly meetings of the annotators, where problems had been discussed and solutions immediately applied to the annotation process. Later, a dedicated editor was chosen from the annotators, and also the technical issues warranted another two annotators to stop annotating and cover the technical area.

The morphological annotation has been performed twice followed by the usual adjudication phase (by a single person to ensure high consistency). The annotators were choosing among possible lemmas and tags offered by the Czech morphological dictionary without any automatic pre-tagging or another kind of preference of tags. Almost 2 mil. tokens have been annotated at the morphological layer manually.

The analytical-layer annotation was performed only once, but with an extensive set of automatic consistency checks that included cross-layer annotation checking. At the beginning, no automatic preprocessing was taking place; later, a hand-written code was used to pre-assign the dependency functions. In 1998, a pre-release called PDT 0.5 was put together (containing about 380k annotated words) for the summer JHU Language Engineering Workshop in Baltimore, MD, U.S.A., where the first Czech parser was developed (by converting the data for the—slightly adapted—Collins' lexicalized English parser). Since 1999, the data for annotation have been preprocessed by this parser and presented to the annotators for corrections only, gaining approx. 30% annotation speedup. Over 1.5 mil. tokens have been manually annotated at the analytical layer, matching the Penn Treebank in size.

Merging the two layers of annotation, a non-trivial task, took over a year. It included extensively checking the data for consistency, final editing of the guidelines (and their translation to English), and finally preparing the CD-ROM for publication in 2001 as the Prague Dependency Treebank, version 1.0. During the checking phase, a new platform-independent editing tool, `TrEd`, has been developed.

The tectogrammatical layer annotation (using `TrEd`), starting in 2000 with the establishment of the Center for Computational Linguistics after the original funding expired, was originally thought to be too difficult to cover all of the planned data (about 50k sentences, a subset of the PDT 1.0 data) in full. The annotation was divided into four areas:

- dependency structure in the form of a dependency tree, including semantic labeling and valency annotation,
- topic/focus annotation,
- coreference (grammatical and a restricted subset of textual one),
- grammatical attributes of the nodes of the tree (not covered by any of the above).

Most of the effort has been directed to the first area, since the others should have been covered by small samples only. Manually written rules have been applied to the analytical-layer trees to pre-annotate them in cases where the relation between the analytical and the tectogrammatical trees was thought to be clear. Rudimentary valency dictionary has been prepared (in a hard-copy form) to assure consistency at least for the annotation around the most frequent verbs. The XML version of the valency dictionary, *PDT-VALLEX*, has been created later and an interface added to `TrEd` allowing for on-line use and editing of the dictionary; it also enabled to assign the appropriate valency frame to an occurrence of a word in the corpus. Meanwhile, as the work on the guidelines and test annotation of coreference and the topic/focus annotation progressed, it has been eventually decided to perform these annotations on the whole data. Still later, in 2004, the fourth area (assignment of additional grammatical information, filling no less than another 16 attributes of every tectogrammatical tree node) was also semi-automatically extended to the whole tectogrammaticaly annotated data, i.e. 50k sentences.

Contrary to the analytical layer of annotation, the tectogrammatical annotation staff has been divided into many small teams, with specialized (sub)areas assigned to their members. This has been a disadvantage, too—information sometimes did not get to all the people for whom it was relevant. Up to 30 people worked on the project at any given time. Everything has been annotated only once, except in pilot inter-annotator agreement tests. Consistency checking has been applied in a similar way as it was to the analytical layer, using complex cross-layer checking.

The final stage (after the "assembly-line" annotation process had finished in 2004) took also over a year. Completely new stand-off XML annotation scheme has been developed for the distribution of the data. The valency dictionary PDT-VALLEX has been fully manually checked and revised for verbs and certain categories of nouns (in both cases, by a single person to ensure maximum consistency), and extensive automatic cross-layer checks have been developed to find annotation inconsistencies—after it, all of them have then been manually corrected. A dedicated editor of the tectogrammatical annotator guidelines was appointed, whose task was to rewrite the individual sections of the guidelines (over 800 pages total) in a clear manner that uses consistent terminology and corresponds to what has eventually been annotated in the data. The guidelines have also been translated into English. The CD-ROM has been completed and shipped to LDC for publication in 2006.

1.4 About Czech

Czech, the language of texts incorporated in the Prague Dependency Treebank, is one of the western group of Slavic languages. It is spoken mainly in the Czech Republic where it is the only official language. Besides, native Czech speakers live in the other European countries, especially in Slovakia, and tens of thousands of Czech speakers live in the U.S.A., Canada and Australia. Czech has over 10 million speakers.

Similarly to other Slavic languages, Czech is highly inflectional—it has seven cases and four genders (e.g. there are 16 main paradigms for inflection of nouns) and it has free word order (from the purely syntactic point of view): words in a sentence can usually be ordered in several ways. However, the particular word order does influence the meaning of the sentence.

Czech is written using the Latin alphabet extended with several letters with accents. Czech letters (82 characters total) are included in the Unicode standard; also ISO 8859-2 (Latin 2), the standard 8-bit encoding for Central-European languages, and CP1250, its Windows counterpart, are widely used.

More information about the Czech language can be found at <`http://www.czech-language.cz`>.

1.5 Directory structure

This section contains a short description of the directory structure of the PDT 2.0 distribution, down to the second level.

- `data/` – see Chapter 3, "Data"
    - `binary/` – all annotated data (on distribution CD-ROM only; see Section 3.6) in Perl Storable Format (see Section 3.4.2)
    - `filelists/` – several pre-generated lists of data files (on distribution CD-ROM only), see Section 3.6

  - `full/` – all annotated data (on distribution CD-ROM only; see Section 3.6) in PML Format (see Section 3.4.1)
  - `pdt-vallex/` – PDT-VALLEX, the valency lexicon, see Section 3.8
  - `pdt1.0-update/` – update of data from PDT 1.0 CD-ROM (on distribution CD-ROM only), see Section 3.9
  - `sample/` – a small sample of annotated data, see Section 3.7
  - `schemas/` – PML and RelaxNG schemata of the data

- `doc/` – see Chapter 5, "Documentation"
  - `data-formats/` – documentation of the data, see Section 3.4
  - `manuals/` – manuals (guidelines) for annotators, see Chapter 2, "Layers of annotation"
  - `pdt-guide/` – this PDT guide
  - `styles/` – cascading stylesheets for manuals and PDT guide
  - `tools/` – documentation of the distributed tools, see Chapter 4, "Tools"
- `publications/` – publications related to PDT 2.0, see Chapter 6, "Publications"
- `tools/` – see Chapter 4, "Tools"
  - `checks/` – macros for detection of errors in data, see Section 4.7
  - `format-conversions/` – conversion utilities among various data formats, see Section 4.4
  - `machine-annotation/` – tools which build syntactic trees over plain Czech sentences, see Section 4.5
  - `netgraph/` – `Netgraph`, a tool for searching in data, see Section 4.1
  - `pml/` – Relax NG definition of PML schema and a XSLT stylesheet for converting PML schemas to RelaxNG, see Section Tools in the PML specification.
  - `tred/` – `TrEd` and `btred/ntred`, tools for viewing and processing data, see Section 4.2, Section 4.3
- `visual-data/`
  - `pdt-vallex/` – PDT-VALLEX, the valency lexicon, as web pages, see Section 3.8
  - `sample/` – sample data as web pages, see Section 3.7

Chapter 2

Layers of annotation

The data in PDT 2.0 are annotated on three layers—the *morphological layer* (Section 2.1), *analytical layer* (Section 2.2), and *tectogrammatical layer* (Section 2.3). In fact, there is also one non-annotation layer, representing the “raw-text”. On this layer, called *word layer*, the text is segmented into documents and paragraphs and individual tokens are recognized and associated with unique identifiers.

The word layer is also called the *w-layer*, the morphological one the *m-layer*, the analytical one the *a-layer*, and the tectogrammatical one the *t-layer*. Similarly, a node of a tree expressing analytical annotation of a sentence is called the *a-node* etc.

Figure 2.1 shows the relations between the neighboring layers as annotated and represented in the data. The rendered Czech sentence *Byl by šel dolesa.* (lit.: *He-was would went toforest.*) contains past conditional of the verb *jít* (*to go*) and a typo.

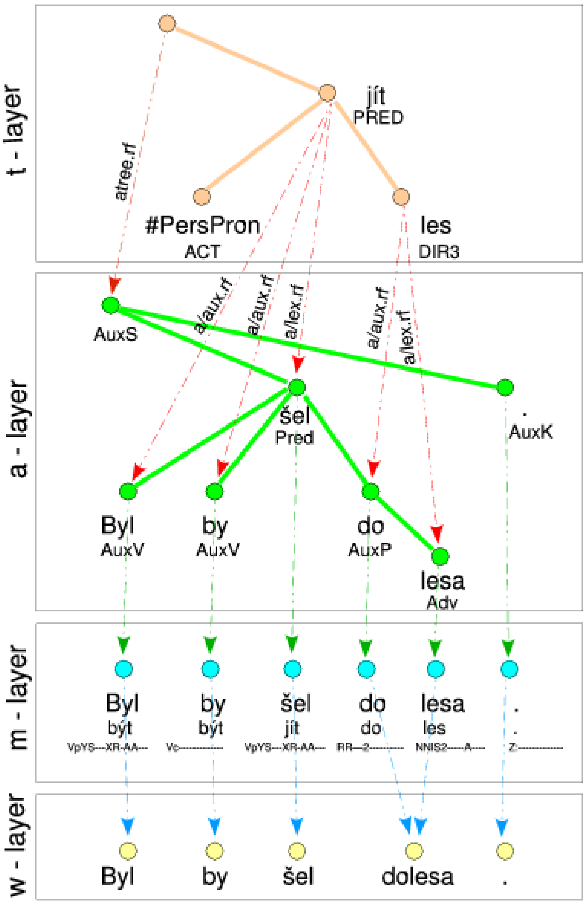


Figure 2.1: Linking the layers

2.1 Morphological layer

This section briefly describes the morphological layer. For more information see the Manual for Morphological Annotation.

2.1.1 Logical structure

At the morphological layer, the sequence of tokens of the w-layer is divided into sentences. Annotation of a sentence consists of attaching several attributes to the tokens of the w-layer, the most important of which are morphological lemma and tag.

2.1.2 Physical realization

Attribute `lemma` carries the lemma of the token. It represents its basic or normalized form, and it matches the unique key of the corresponding entry in the morphological dictionary. The `tag` attribute contains 15-character long morphological tag that expresses the part of speech of the token and its various morphological categories. The attribute `id` contains a PDT 2.0-unique identifier of the m-layer unit which is then used for back reference from the analytical layer (for the overall linking scheme, see Figure 2.1), and the reference-type `w.rf` attribute refers back to the w-layer. Several other attributes handle possible (but rare) corrections and/or normalizations relative to the w-layer; the most important of them is the `form` attribute which contains the correct form of the text token (which might differ from the text token in case of spelling errors, incorrectly split or joined words, unsuitable representation of decimal points in numbers or of other technical problems with the original text).

See a sample sentence in Table 2.2

2.1.3 Annotation process

The morphological layer of PDT has been annotated by a group of seven annotators. The group proceeded in two separate phases. During the first phase—after each text had been processed by the automatic morphological analyzer—two annotators independently chose the lemma and the morphological tag from the list suggested by the morphological analyzer. In the second (adjudication) phase, an annotator-arbiter resolved the differences between them.

After the separate checking of the morphological and syntactical-analytical layers, a mutual revision was done. The comparison concentrated on the aspects of analytical functions vs. morphological tags, preposition vs. the case of depending node, and finally, the "case, gender, number" agreement between a dependent and its governing node.

2.2 Analytical layer

This section briefly describes the analytical layer. For more information see the Annotations at Analytical Level.

2.2.1 Logical structure

A sentence at the analytical layer is represented as a rooted ordered tree with labeled nodes and edges. One token of the morphological layer (see Section 2.1) is represented by exactly one node of the tree and the dependency relation between two nodes is captured by an edge between the two nodes. The actual type of the relation is given as a function label of the edge. Most of the edges represent dependency relations, while others mirror various linguistic or technical phenomena, e.g. coordination, apposition, punctuation, etc. Linear ordering of the nodes, which corresponds to the original sentence word order, is also recorded, allowing "correct" graphical rendering of the tree.

2.2.2 Physical realization

A set of six attributes is attached to every node (except for the technical root of the tree that has less attributes). The attribute `id` contains a PDT 2.0-unique identifier of the node which is referred back (linked) from the tectogrammatical layer (see Figure 2.1). The linear-order attribute `ord` contains the corresponding token position in the original sentence. For technical simplicity, the analytical function attribute `afun` belonging to an edge is moved to the dependent end of the edge and appears as a node attribute. The attributes `is_member` and `is_parenthesis_root` mark proper coordination, apposition and parenthesis interpretation. Finally, the attribute `m.rf` links the node to the corresponding morphological one.

See a sample tree in Figure 2.3

2.2.3 Annotation process

All the analytical data have been annotated manually by a team of six annotators. At the beginning, the annotators had to build the whole tree and assign all the analytical functions, while in the later stages, sentences were pre-processed by a parser and a tree was proposed to the annotators. A rule-based function assigning analytical functions was available as well. Nevertheless, the annotators had to revise the output of both the procedures since the result of these procedures was often inacurate.

After the annotation process had finished, the data passed many checking tests. An example of such a test was verification of the assertion that the verbal nominal predicate (indicated by analytical function `Pnom`) must always have the verb *být* (*to be*) as its head. All the violations of these rules/tests were manually checked and corrected.

2.3 Tectogrammatical layer

This section briefly describes the tectogrammatical layer. For more information see the Tectogrammatical Annotation of the PDT: Annotator's Guidelines.

2.3.1 Logical structure

The tectogrammatical representation of a sentence captures the following aspects:

- **Tectogrammatical structure and functors.** Every sentence is represented as a rooted tree with labeled nodes and edges. The tree reflects the underlying (deep) structure of the sentence. The nodes stand for auto-semantic words only (with some exceptions of a technical nature). Unlike the analytical layer, not all the morphological tokens are represented at the tectogrammatical layer as nodes (for example, there are no prepositions on the tectogrammatical layer) and some of the tectogrammatical nodes do not correspond to any morphological token (for example, the structure contains a node representing omitted subject in pro-drop constructions). *Grammatemes* are attached to some nodes; they provide information about the node that cannot be derived from the structure, the functor and other attributes (for example number for nouns, modality and tense for verbs etc.). The edges of the tree represent relations between the nodes they connect; the type of the relation is indicated by the label of a particular edge (similarly to the analytical layer). Every node representing a verb or a certain type of a noun has a valency frame assigned to it (by means of a reference to a valency dictionary entry—see Section 3.8).

- **Topic–focus articulation (TFA).** Each node is assigned one of the three values assigned on the basis of contextual boundness: a node can be contextually bound, contrastively contextually bound, or contextually non-bound. In addition, the nodes in the topic part of the sentence are ordered according to the assumed communicative dynamism.

- **Coreference.** At the current phase of annotation, coreference relations between nodes of certain category types are captured, distinguishing also the type of the relation (textual, grammatical, or the "second dependency" of complement).

2.3.2 Physical realization

The total of 39 attributes is assigned to every non-root node of the tectogrammatical tree; based on the node type (attribute `nodetype`), only a certain subset of the attributes is necessarily filled in. Often, the attributes are of a list or set type, containing more than one value.

- **Tectogrammatical structure and functors.** Similarly to the analytical layer, a set of attributes is attached to every node; however, there are many more attributes at the tectogrammatical layer for the description of the linguistic structure than at the analytical one. The attribute `id` contains a unique identifier of the node, the attribute `functor` describes the type of the edge leading from the node to its governor (the edge may represent dependency relation or other technical phenomena). The attribute `t_lemma` stands for the tectogrammatical lemma of the node. Grammatemes are rendered as a set of 16 attributes grouped by the "prefix" `gram` (e.g. `gram/verbmod` for verbal modality). Attributes for back reference (linking) to the analytical layer are provided (see Figure 2.1), as well as other attributes for coordination and apposition, parenthesis, direct speech, quotations, etc.

- **Topic–focus articulation.** Classification of nodes as contextually bound, contrastively contextually bound, or contextually non-bound is represented by the corresponding value of attribute `tfa`. The numeric attribute `deepord` is used for the underlying ordering of nodes based on communicative dynamism.
- **Coreference.** Attributes `coref_text.rf`, `coref_gram.rf`, and `compl.rf` contain `id`s of coreferential nodes of the respective types. Attribute `coref_special` contains information about special cases of coreference.

See a sample tree in Figure 2.4

2.3.3 Annotation process

As the tectogrammatical structure is also based on the relation of dependency, automated procedures were used to convert dependency-based analytical trees to an intermediate tectogrammatical-like ones. All the generated intermediate trees were then processed by human annotators, who added a great amount of missing information and corrected the wrong one. Coreference, topic–focus articulation, and some of the grammatemes were annotated separately. All the data were then checked by many post-annotation tests (see Section 4.7).

In Figure 2.2 the data and work flow diagram is shown. Thick arrows represent iterated operation, double arrows indicate merging procedures that were used when different sub-layers were being annotated on the same data at the same time.

2.4 Sample preview of annotation on the three layers

Table 2.1: An example sentence

Některé	*kontury*	*problému*	*se*	*však*	*po*	*oživením*
Some	*contours*	*problem* (gen)	reflexive pronoun	*though*	*after*	*resurgence* (instr)

Some contours of the problem seem to be clearer after the resurgence by Havel's speech.

Havlovým	*projevem*	*zdají*	*být*	*jasnější*	.
Havel's	*speech* (instr)	*they-seem*	*to-be*	*clearer*	.

An example sentence can bee seen in Table 2.1.

The annotation of the sample sentence on the morphological layer can bee seen in Table 2.2. Note that the instrumental form of *oživení* was changed to locative form. The reason (indicated in `form_change` element) is a spelling error.

Annotation of the sample sentence on the analytical layer can be seen in Figure 2.3. Note that the word *zdají* is labeled as the only coordination member. This is how a coordination with the previous sentence has been annotated on analytical layer.

The annotation of the sample sentence on the tectogrammatical layer is in Figure 2.4.

Note that the word *však* is no longer a coordination node. It was labeled by the functor `PREC` as a linking word to the previous sentence. Also note that the word *se* became part of the complex verb form *zdát_se*, that the preposition *po* disappeared (but it is referred to from the word *oživení* and it is the basis of the functor and sub-functor values of this word), that the pronoun *některý* has `t_lemma` *který* but its indefiniteness is expressed by the value of `gram/sempos` and `gram/indeftype`, etc.

For more examples, see Section 3.7.

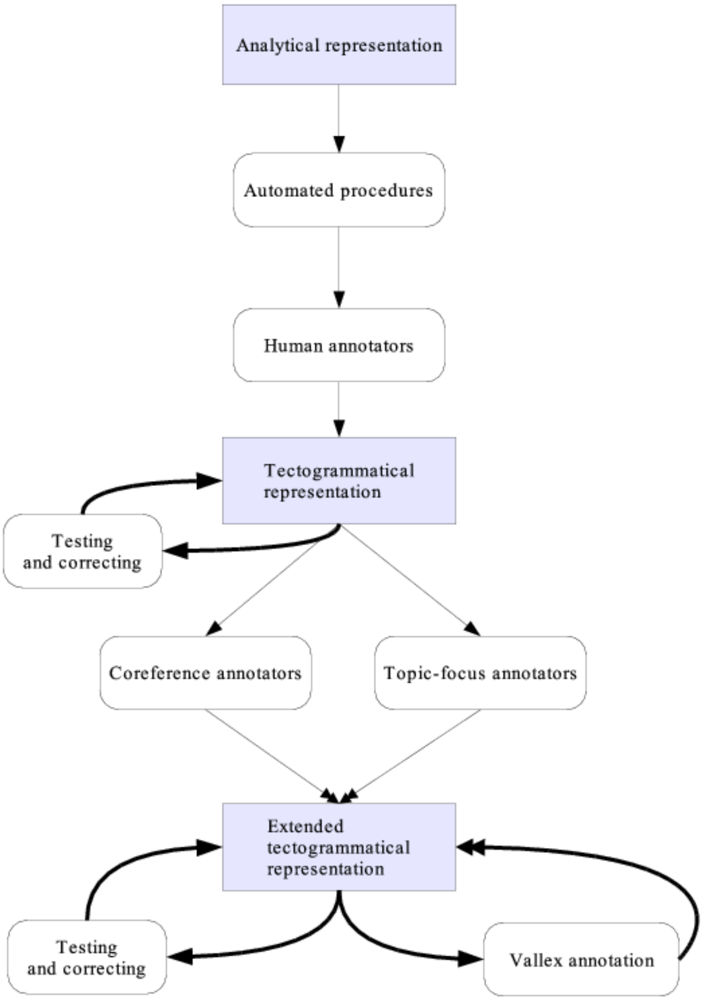


Figure 2.2: Data and annotation workflow diagram

Table 2.2: Morphological analysis of the example sentence

Form	Lemma	Morphological tag
Některé	*některý*	`PZFP1----------`
kontury	*kontura*	`NNFP1-----A----`
problému	*problém*	`NNIS2-----A----`
se	*se*_^(zvr._zájmeno/částice)	`P7-X4----------`
však	*však*	`J^-------------`
po	*po*-1	`RR--6----------`
oživení	*oživení*_^(*3it)	`NNNS6-----A----`
Havlovým	*Havlův*_;S_^(*3el)	`AUIS7M---------`
projevem	*projev*	`NNIS7-----A----`
zdají	*zdát*	`VB-P---3P-AA---`
být	*být*	`Vf--------A----`
jasnější	*jasný*	`AAFP1----2A----`
.	.	`Z:-------------`

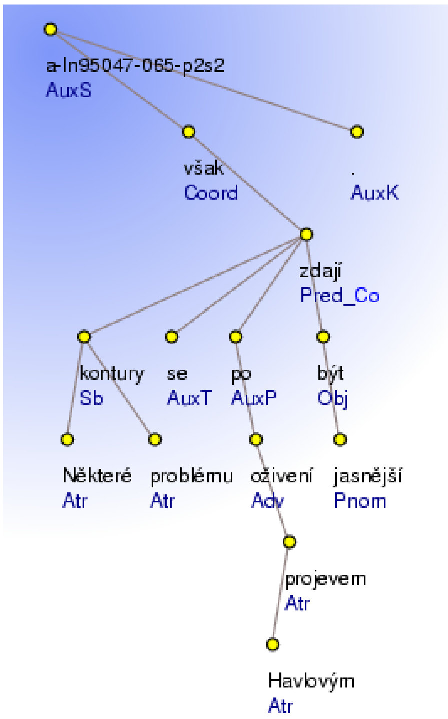


Figure 2.3: The analytical tree of the example sentence

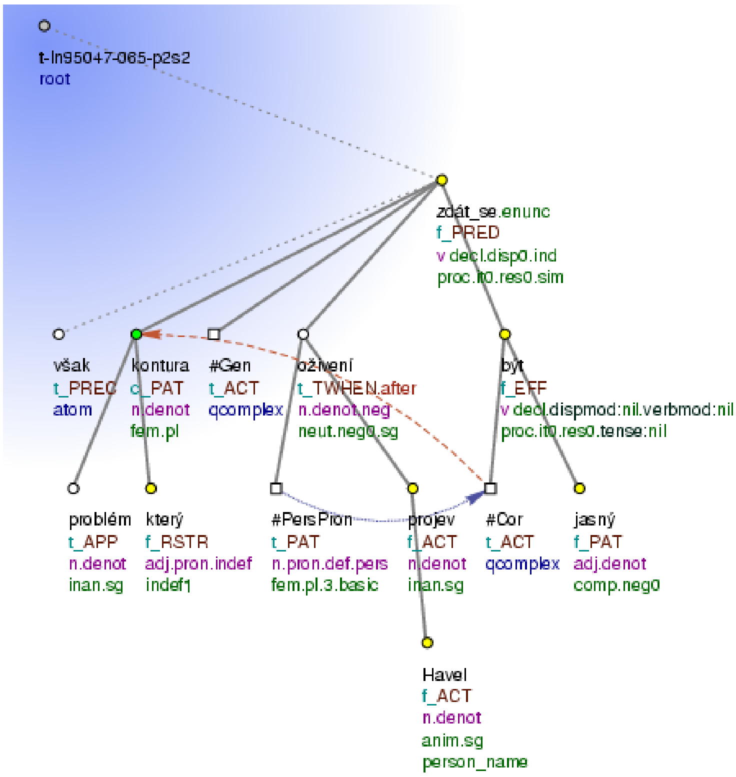


Figure 2.4: The tectogrammatical tree of the example sentence (a detailed view)

Chapter 3

Data

The data is the only part of PDT 2.0 distribution which cannot be downloaded from the website of PDT, <http://ufal.mff.cuni.cz/pdt2.0/>. The only downloadable parts of the data are sample data (see Section 3.7) and PDT-vallex (see Section 3.8). If you wish to obtain also the full data (see Section 3.6) and PDT 1.0 update (see Section 3.9), you have to get the distribution CD-ROM—Chapter 7, "Distribution and license" describes how to do it.

The data are located in the directory `data`.

3.1 Sources of text

The data in Prague Dependency Treebank are annotated articles (non-abbreviated) from the following newspapers and journals:

- Lidové noviny[1] (daily newspapers), ISSN 1213-1385, 1991, 1994, 1995
- Mladá fronta Dnes[2] (daily newspapers), 1992
- Českomoravský Profit[3] (business weekly), 1994
- Vesmír[4] (scientific journal), ISSN 1214-4029, Vesmír, s.r.o., 1992, 1993

The amount of data from the particular sources is given in Figure 3.1.

The texts in electronic form have been provided by the Institute of the Czech National Corpus.[5] The texts came from their providers in several formats. Sometimes original formating has been preserved but in general only the division to documents (articles) and paragraphs has been adopted.

For various reasons (mostly just mistakes), the original data contained duplicates. When a duplicity was longer than three sentences, it has been removed. Further, very high frequency non-word data like over-typings of chess games, tables with results of sport matches etc. have almost all been removed with a few kept to remind us of their existence and to show a suggested (rather technical) annotation scheme for them.

3.2 Division of the data according to the layer of annotation

Annotations of the particular layers do not cover the data equally. The more complex a layer is, the less data have been annotated at it. The reason is obvious—annotation of a more complex layer needs more time, resources, and human work; there are other technological considerations as well (e.g., for a certain setup of higher-layer tool development, there must be more data available for training purposes on the lower layer the annotation of which cannot be used at the upper layer anyway). Any file annotated at a certain layer *is annotated also* at all the simpler ones. See Figure 3.2 for an illustration.

For details on layers, see Chapter 2, "Layers of annotation". For details on reflecting layers of annotation in names of files, see Section 3.5. For details on data quantities, see Section 3.6.

[1] <http://lidovky.centrum.cz/archivln/>

[2] <http://zpravy.idnes.cz/mfdnes.asp>

[3] <http://www.profit.cz/>

[4] <http://www.vesmir.cz/>

[5] <http://ucnk.ff.cuni.cz/>

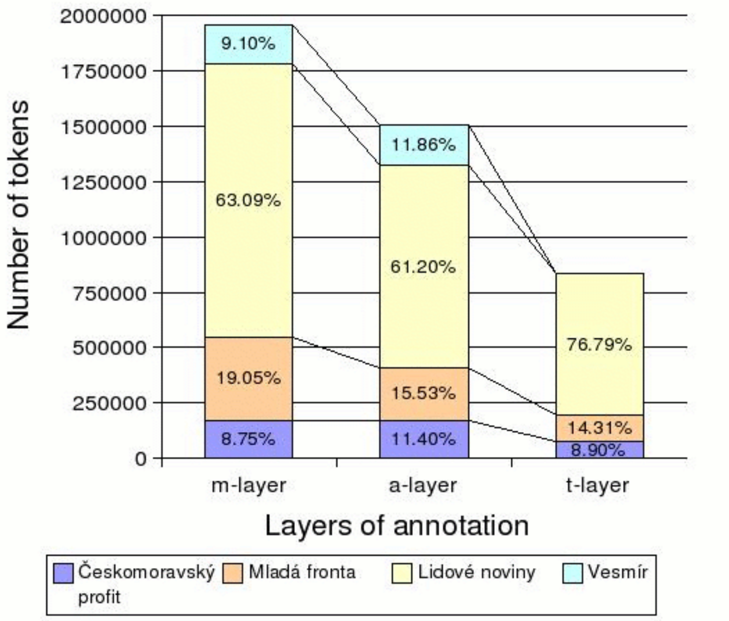


Figure 3.1: Number of tokens from the particular sources

3.3 Division of the data into training and test sets

As usual, the data are divided into three groups: the training data, the development test data and the evaluation test data. The training data cover approximately 80%, development 10% and evaluation 10% of the whole set of data (these proportions hold for all the three layers of annotation).

The users can freely exploit the training set and test their hypotheses or tools on the development test data. Evaluation test data should be never looked into, they are intended for evaluation and reporting purposes *only*. Moreover, the evaluation data should be used advisedly and as rarely as possible, since the observations gained from the repeated tests on the evaluation data could lead to a change of the original hypothesis/tool and thus the evaluation data would start functioning as the training data.

Although the train/dtest/etest proportion is roughly the same as in PDT 1.0 (8:1:1), the old division has not been preserved due to several reasons. The data in PDT 2.0 were divided in the following way: documents of the morphological layer were read in sequence and cyclically distributed, the first one was folded into *train-1* set, second one into *train-2*, and so on to *train-8*, the ninth to *dtest* and tenth to *etest*. Eleventh document went to *train-1* again etc. (Training set was divided into eight subsets to lower the number of files in directories but the existence of ten equally large sets of data might serve in cross-validation experiments as well.) Documents annotated on the other layers went together with their morphologically annotated versions. Since the documents for annotation were selected sequentially, the algorithm guarantees that the proportions remain almost the same (8:1:1), with only a small deviation due to differences in the size of the documents.

Figure 3.3 shows the division of the data. Note that the algorithm makes sure that every file belongs to the same set (training vs. development test vs. evaluation test) on all the layers it has been annotated at. (For details on data quantities, see Section 3.6.)

It should be noted that if the user performs for instance an experiment on a-layer data and the experiment has nothing to do with t-layer, then s/he should use such division of the data which disregards the fact whether the document in question is annotated on t-layer or not. As a result, e.g. *etest* subset of the a-layer data is in fact composed of two parts, as it is visible in Figure 3.3 (two vertically shaded areas in the middle row). By analogy, *train-1* subset of the m-layer data is composed of three parts. The issues related to such groupings are also addressed in Section 3.6.

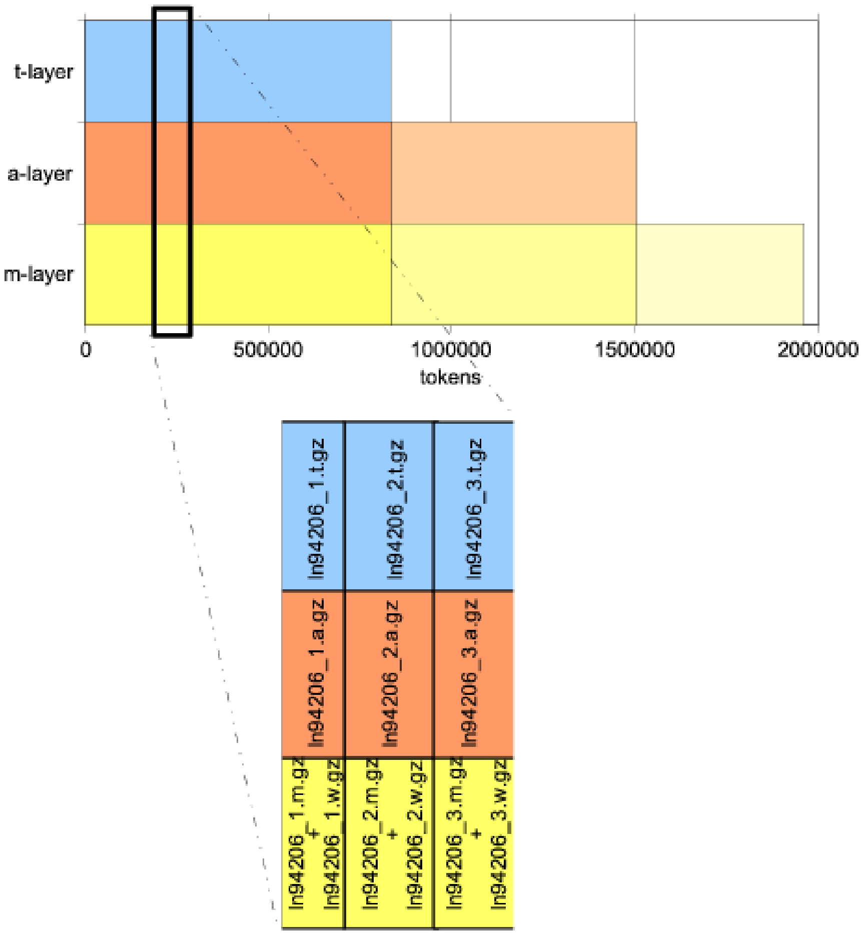


Figure 3.2: Division of the data to layers

3.4 Data formats

The primary data format for PDT 2.0 is an XML[6]-based format called *PML*. Historically, two other formats have been developed and used for processing and storage of PDT data. The *FS* format has been developed for `Netgraph` program (strictly speaking, for its ancestor, `Graph` program). A SGML-based format, called *CSTS*, has been the primary format of PDT 1.0. It is now used only as an intermediate format in older NLP tools (such as taggers and parsers).

For information on conversion between these formats, see Section 4.4.1.

3.4.1 PML

PML ("Prague Markup Language"), is a generic XML-based data format designed for the representation of rich linguistic annotation of text, such as morphological tagging, dependency trees, etc. PML is an on-going project in its early stage. Yet, enough has already been developed to allow an adequate and straightforward representation of the PDT 2.0 data. In the following text, we give a brief summary of PML main features. A detailed information about PML as a generic format can be found in the PML documentation. An overview of how PDT 2.0 data are actually represented in PML can be found in the PDT 2.0 Annotation Markup Reference.

In PML, individual layers of annotation can be stacked one over another in a stand-off fashion and linked together as well as with other data resources in a consistent way. Each layer of annotation is described in a *PML schema* file, which could be imagined as a formalization of an abstract annotation scheme for the particular layer of annotation. In brief, the PML schema file describes which elements

[6] <`http://www.w3.org/XML/`>

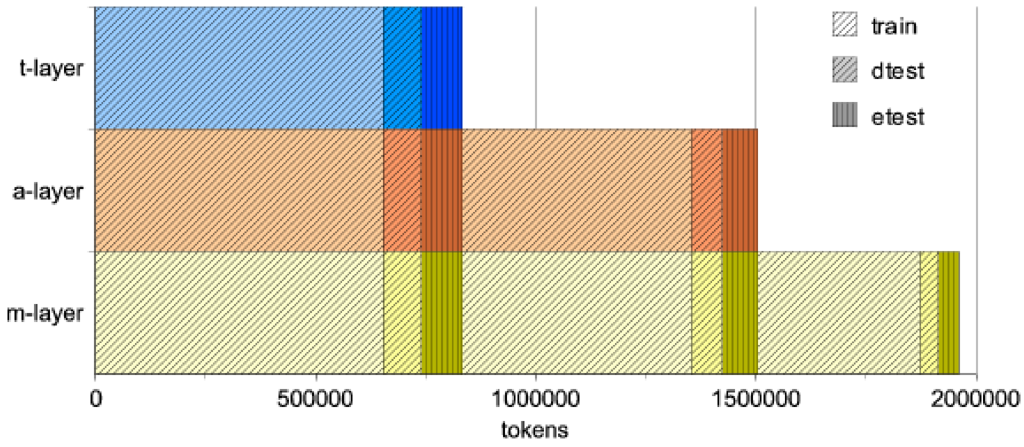


Figure 3.3: Division of the data into training and test sets

occur on that layer, how they are nested and structured, of which types the values occurring in them are, and what role they play in the annotation scheme (this *PML-role* information can also be used by applications to determine an adequate way to present a PML instance to the user). Other formal schemata such as Relax NG[7] can be automatically generated from a PML schema, so that formal consistency of PML-schema instances could be verified using conventional XML-oriented tools (a XSLT stylesheet providing conversion of PML schema to Relax NG is available in tools/pml/pml2rng.xsl).

Every PML instance starts with a header where a PML schema is associated with the instance and where all external resources which the instance points to are listed, together with some additional information necessary for correct link resolving. The rest of the instance is dedicated to the annotation itself.

The annotation is expressed by means of XML elements and attributes, named and nested according to the associated PML schema. XML elements of a PML instance occupy a dedicated namespace `http://ufal.mff.cuni.cz/pdt/pml/`. PML format offers unified representations for the most common annotation constructs, such as attribute-value structures, lists or alternatives of values of a certain type (either atomic or further structured), references within a PML instance, links among various PML instances (used in PDT 2.0 to create links across layers) or to other external XML-based resources. At the moment, PML also offers a limited support for XML mixed content. To avoid confusion with XML attributes, we usually refer to attributes of an attribute-value structure as *members*.

PDT 2.0 contains annotation divided into up to four layers stacked one upon another, namely the word, morphological, analytical, and tectogrammatical layers (see Chapter 2, “Layers of annotation”). Each of the layers defines its own PML schema.

Tectogrammatical and analytical trees are represented in PML commonly as nested attribute-value structures. In this representation, a node is realized as an attribute-value structure with PML-role `#NODE`. Each node has a dedicated member with a PML-role `#CHILDNODES`, which contains a list of child-nodes of the node. Because of the auxiliary character of root nodes of the dependency trees of PDT 2.0, the structure representing the technical root of the tree is of a type different from the rest of the nodes (i.e. has a different set of members).

See also the PDT 2.0 Annotation Markup Reference for a comprehensive overview of the PML representation of the four annotation layers. PML and Relax NG schemata for the four layers can be found in the directory `data/schemas`.

3.4.2 Perl Storable Format

When working with PML, which is the XML-based primary format of PDT 2.0, the tools based on Perl such as `TrEd` and `btred` parse the original XML and build their internal memory representation. This transformation is time consuming, but can be completely avoided when working with the `pls.gz` data format (Perl Storable Format). It is a binary format which directly mirrors the internal memory representation and thus is much faster to store and load. But on the other hand, this format has nothing to do with XML any more and is hardly processible by other tools.

[7] <http://www.relaxng.org/>

3.4.3 FS

The FS ("feature structure") file format is a generic format for representation of trees whose nodes are attribute-value structures. It can be viewed as a "meta-format", similarly to SGML or XML. An application of this format is fully specified by attribute declarations in a FS-file header (thus with respect to FS format, the header of an FS-file plays a similar role to that of DTD with respect to a particular application of SGML).

Every FS-file starts with a declaration of its attributes. In general, each line of the declaration consists of `@`-character, property of the attribute, a space, and name of that attribute. E.g. property `O` means "obligatory", i.e. values of such an attribute must be non-empty for every node; or property `L`, "list", requires that the attribute value (if not empty) is one of those listed in a `|`-separated list following the attribute name. The complete description can be found in the FS format specification.

The declaration header ends with an empty line and it is followed by descriptions of trees representing the annotation. Every tree description starts on a new line. Trees are described in a usual parentheses notation, i.e. after the description of a node the list of its child-nodes enclosed in parentheses follows. Descriptions of individual child-nodes are separated by commas. Description of every node is enclosed in square brackets and consists of comma-separated list of *`attribute=value`* pairs. When an attribute is declared as `P`, "positional", it can be given only by its value and its name is derived as the name of the first positional attribute whose definition in the header follows the definition of the last attribute in the list (or the first positional attribute if the value is the first attribute description occurring in the list).

3.4.4 CSTS

CSTS ("Czech sentence tree structure") is an application of SGML. CSTS has been the primary format for PDT 1.0 and in spite the fact that in PDT 2.0 it has been superseded by PML, some tools still depend on it. CSTS can only represent morphological and analytical annotation (to be precise, its definition contains also some elements related to tectogrammatical annotation, but it is not capable of fully capturing the t-layer of PDT 2.0). Wherever possible, we highly recommend using PML (see Section 3.4.1) instead—this applies especially to any new tools. For more details, see complete description of CSTS and its DTD file.

3.5 Conventions of file naming

The data of PDT 2.0 are distributed in the PML format (see its description in Section 3.4.1). Each data file relates to one annotated document—the base of its name is the identifier of the document (and it indicates the source of the document, see Section 3.1: `ln*` denotes Lidové noviny, `mf*` denotes Mladá fronta Dnes, `vesm*` denotes Vesmír, and `cmpr*` denotes Českomoravský profit). The extension of the file expresses the layer of annotation of the document (`.w` denotes w-layer, `.m` denotes m-layer, `.a` denotes a-layer, and `.t` denotes t-layer). (See the description of the layers in Chapter 2, "Layers of annotation".)

Every file with annotation of a document at some layer relates one-to-one to files with its lower-layer annotations and contains links into them. This is the reason why the files should not be renamed. Links from lower layers to higher layers of annotation do not exist. For an overview of layer linking, see also Figure 2.1.

For example, `cmpr9406_001.a.gz` denotes (gzipped) file with a-layer of annotation of document *cmpr9406_001* (originating from Českomoravský profit). It contains links into files `cmpr9406_001.m.gz` and `cmpr9406_001.w.gz`; however, it says nothing about the existence of file `cmpr9406_001.t.gz`.

Whether a file is a part of the training set or the test one etc. is not captured in its name but with its place in a directory structure, see Section 3.3.

Names of identifiers of sentences and tokens are derived from the name of the file they occur in. Every identifier is unique in the whole treebank.

3.6 Full data

The full version of the PDT 2.0 data is available to the licensed users who obtained CD-ROM PDT 2.0 from Linguistic Data Consortium (see Chapter 7, "Distribution and license"). Small data sample can also be downloaded from the web (see Section 3.7).

The full version of the PDT 2.0 data consists of 7,110 manually annotated textual documents, containing altogether 115,844 sentences with 1,957,247 tokens (word forms and punctuation marks). All these documents are annotated on the m-layer. 75% of the m-layer data are annotated on the a-layer (5,330 docs., 87,913 sents., 1,503,739 toks.). 59% of the a-layer data are annotated also on the t-layer (i.e. 45% of the m-layer data; 3,165 docs., 49,431 sents., 833,195 toks.).

The full data are located in the directory data/full on the CD-ROM PDT 2.0. (In parallel, the full data annotated at least on the a-layer are—merely for the benefit of faster processing by TrEd-based tools—converted also into the binary Perl Storable Format; the converted files are to be found in the directories data/binary/amw and data/binary/tamw.) The data files are divided according to the following two-level hierarchy:

- The primary branching corresponds to the highest layer of annotation (see Chapter 2, "Layers of annotation") available for the document in question:
  - data/full/tamw/ – documents annotated on all three layers,
  - data/full/amw/ – documents annotated only on the m-layer and a-layer,
  - data/full/mw/ – documents annotated only on the m-layer.
- Then, the content of each of these three directories is further split into ten parts of roughly equal size (see Section 3.3). Eight of them are to be used for training purposes (from train-1/ to train-8/), one for development tests (dtest/) and one for evaluation tests (etest/).

Even if the data files are distributed into as many as thirty directories, the amount of files in individual directories still remains large. This is partially due to the fact that the number of physical files (compared to the number of the original textual documents) is multiplied by the factor of four in case of tamw data (for each document, there are four files containing its annotation on respective layer stored in the same directory, see Section 3.5), by three in amw, and by two in mw. Thus the total number of data files is 4 x 3165 + 3 x 2165 + 2 x 1780 = 22715. For instance, the directory data/full/tamw/train-3/ contains 4 x 317 = 1,268 data files.

Note that no data file occurs twice in data/full/ (e.g., the *.m files from data/full/amw/ do not appear again in data/full/mw/). All the thirty subdirectories have mutually disjoint contents, as they contain annotations of different texts.

Detailed quantitative properties of the data distributed according to the above principles are presented in Table 3.1, Table 3.2, and Table 3.3.

Table 3.1: Data annotated on all three layers (tamw).

tamw	train	dtest	etest	total
Location on the CD-ROM	tamw/train-1/ ... tamw/train-8/	tamw/ dtest/	tamw/ etest/	tamw/*/
# documents	2,533 (80.0 %)	316 (10.0 %)	316 (10.0 %)	3,165 (100.0 %)
# sentences	38,727 (78.3 %)	5,228 (10.6 %)	5,476 (11.1 %)	49,431 (100.0 %)
# tokens	652,544 (78.3 %)	87,988 (10.6 %)	92,663 (11.1 %)	833,195 (100.0 %)

Table 3.2: Data annotated only on m-layer and a-layer (amw).

amw	train	dtest	etest	total
Location on the CD-ROM	amw/train-1/ ... amw/train-8/	amw/ dtest/	amw/ etest/	amw/*/
# documents	1,731 (80.0 %)	217 (10.0 %)	217 (10.0 %)	2,165 (100.0 %)
# sentences	29,768 (77.4 %)	4,042 (10.5 %)	4,672 (12.1 %)	38,482 (100.0 %)
# tokens	518,647 (77.3 %)	70,974 (10.6 %)	80,923 (12.1 %)	670,544 (100.0 %)

Table 3.3: Data annotated only on m-layer (mw).

mw	train	dtest	etest	total
Location on the CD-ROM	mw/train-1/ ... mw/train-8/	mw/dtest/	mw/etest/	mw/*/
# documents	1,422 (79.9 %)	179 (10.1 %)	179 (10.1 %)	1,780 (100.0 %)
# sentences	22,333 (80.0 %)	2,610 (9.3 %)	2,988 (10.7 %)	27,931 (100.0 %)
# tokens	364,640 (80.4 %)	42,689 (9.4 %)	46,179 (10.2 %)	453,508 (100.0 %)

Those who want to work only with the m-layer or a-layer data no matter whether the documents are annotated also on the higher layer(s) or not should use alternative groupings. For instance, when experimenting with all the m-layer data, the training data should consist of all of data/full/{tamw,amw,mw}/train-[1-8]/*m.gz files.

Numbers of all documents annotated on the m-layer (no matter whether a-layer and t-layer annotations exist) are merged in Table 3.4. All documents annotated on the a-layer (no matter whether t-layer annotation exists) are merged in Table 3.5.

Table 3.4: Alternative grouping: All data annotated on m-layer (union of tamw, amw, and mw).

all_m	train	dtest	etest	total
Location on the CD-ROM	*/train-1/ ... */train-8/	*/dtest/	*/etest/	*/*/
# documents	5,686 (80.0 %)	712 (10.0 %)	712 (10.0 %)	7,110 (100.0 %)
# sentences	90,828 (78.4 %)	11,880 (10.3 %)	13,136 (11.3 %)	115,844 (100.0 %)
# tokens	1,535,831 (78.5 %)	201,651 (10.3 %)	219,765 (11.2 %)	1,957,247 (100.0 %)

Table 3.5: Alternative grouping: All data annotated on a-layer (union of tamw and amw).

all_a	train	dtest	etest	total
Location on the CD-ROM	*a*/train-1/ ... *a*/train-8/	*a*/ dtest/	*a*/ etest/	*a*/*/
# documents	4,264 (80.0 %)	533 (10.0 %)	533 (10.0 %)	5,330 (100.0 %)
# sentences	68,495 (77.9 %)	9,270 (10.5 %)	10,148 (11.5 %)	87,913 (100.0 %)
# tokens	1,171,191 (77.9 %)	158,962 (10.6 %)	173,586 (11.5 %)	1,503,739 (100.0 %)

Needless to say that any published experiment performed on the PDT 2.0 data should be accompanied with the information specifying which part of the data was used (for which purpose) in the experiment.

In order to facilitate the work with the large number of data files, we provide the user with pregenerated file lists located as separate files in the directory data/filelists/; not only they are useful when working in tred/btred/ntred environment, but the file-list style of work avoids the problems related to having too many arguments on a command line. However, only a few basic file lists are given, since it is not difficult for the user to create a new file list corresponding to any desired subset of the full data (see also btred/ntred tutorial).

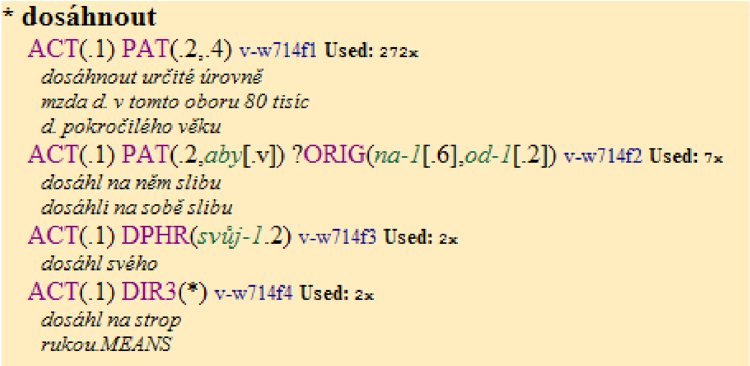
3.7 Sample data

A small portion of the full data is also available from the website for download (again, how to order the full version see in Chapter 7, "Distribution and license"). The data are divided into ten groups (*sample0* to *sample9*) of approximately 50 sentences each. Each group consists of four files (`sampleX.w.gz`, `sampleX.m.gz`, `sampleX.a.gz`, and `sampleX.t.gz`); the extension of a file expresses that the file contains annotation of a sample at the respective layer (see Section 3.5). Sample data are randomly selected segments of the full data (see Section 3.6).

The sample data are stored in the directory `data/sample`. In the same directory, there also is the archive of all the sample files. If you cannot or do not want to install a tool that can deal with the data in PML format (see Chapter 4, "Tools"), you might wish to view all the sample data as web-pages.

3.8 PDT-VALLEX

PDT 2.0 contains a limited lexical semantic annotation that links the underlying and surface syntax and morphology in a novel way by means of a *valency dictionary*, called *PDT-VALLEX*. It is stored in the directory `data/pdt-vallex` in an XML format (see its description) or can be browsed as web-pages—see visualization of its sample entry in Figure 3.4. The entry *dosáhnout* (*to reach*) has the following frames: (1) to reach (a certain level), (2) to make sbd. promise sth., (3) to achieve one's goal, (4) to reach (up to sth.).



* **dosáhnout**
ACT(.1) PAT(.2,.4) v-w714f1 **Used: 272x**
dosáhnout určité úrovně
mzda d. v tomto oboru 80 tisíc
d. pokročilého věku
ACT(.1) PAT(.2,*aby*[.v]) ?ORIG(*na-1*[.6],*od-1*[.2]) v-w714f2 **Used: 7x**
dosáhl na něm slibu
dosáhli na sobě slibu
ACT(.1) DPHR(*svůj-1*.2) v-w714f3 **Used: 2x**
dosáhl svého
ACT(.1) DIR3(*) v-w714f4 **Used: 2x**
dosáhl na strop
rukou.MEANS

Figure 3.4: PDT-VALLEX sample entry in the presentation format

Entries of PDT-VALLEX contain individual *senses* of verbs and certain verbal nouns and adjectives that have been found in the corpus. Each sense contains a *valency frame* with semantic, syntactic and morphological information about its semantically obligatory and/or optional dependents.

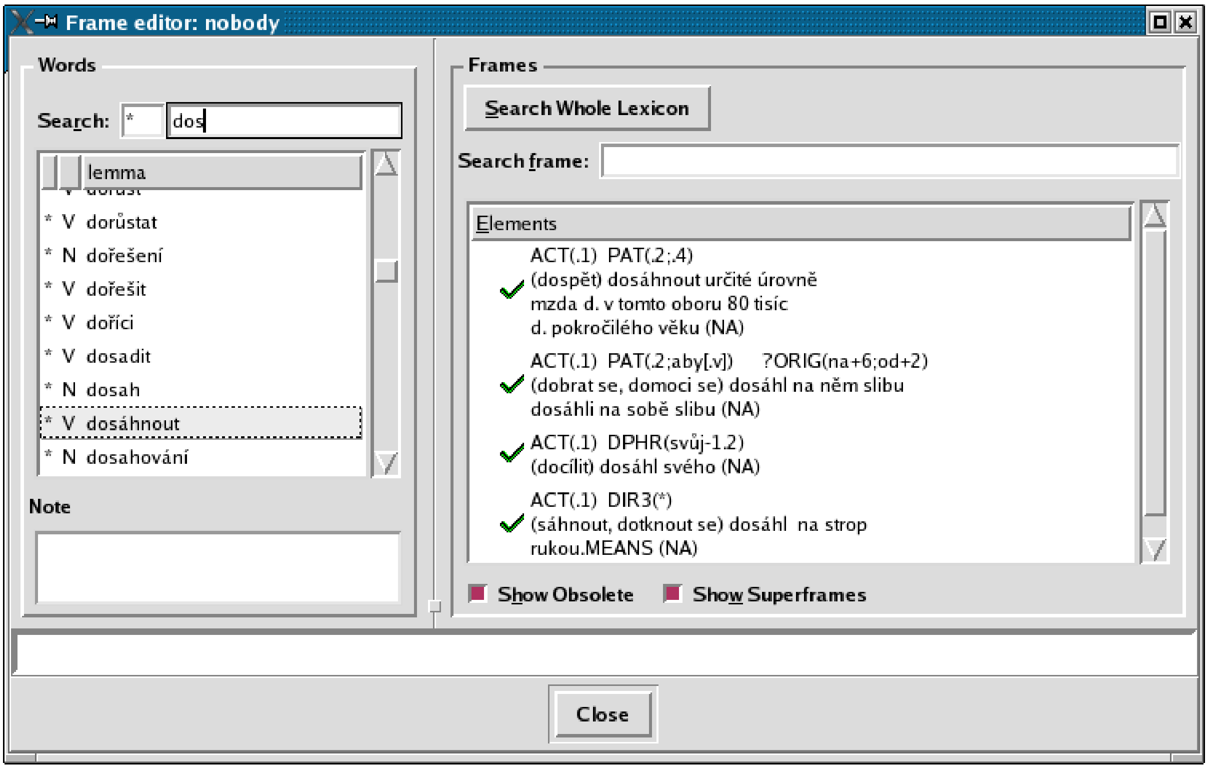
Every valency frame contains zero or more *valency slots*. Each slot has a syntactic or semantic label (such as `ACT`, `PAT`, `ADDR`, `LOC`, `AIM`, `CRIT`, `BEN` etc.; for more about the tectogrammatical annotation in general, see Tectogrammatical Annotation of the PDT: Annotator's Guidelines), and it is marked either as obligatory or optional. In addition, the slots contain surface syntactic and morphological information about their surface realization (expression), such as morphological case, or preposition to be used with the corresponding lexical unit, or, in the case of phrasal expressions, a whole syntactic subtree that forms the phrase on the surface.

The most important feature of PDT-VALLEX is, however, the fact that every occurrence of a verb and a verbal noun in PDT 2.0 is linked (using a special sentence node attribute of a reference type) from the corpus to the dictionary entry, effectively creating a disambiguated *word sense annotation* for these words. The labels, optionality/obligatoriness, and surface morphological form(s) of the entry being pointed to from the corpus have been fully checked against the data annotation at all three layers, as appropriate.

Tools allowing to take advantage of the links between the corpus and the dictionary (simultaneous browsing, searching and editing by the `TrEd` editor—see Figure 3.5) are provided.

3.9 PDT 1.0 update

Although the main difference between PDT 1.0 and PDT 2.0 is the presence of the annotation on the tectogrammatical layer (see Section 2.3), many changes have been done on the lower layers, too. For



Search: * dos

* V dorůstat
* N dořešení
* V dořešit
* V dořící
* V dosadit
* N dosah
* V dosáhnout
* N dosahování

ACT(.1) PAT(.2;.4)
✓ (dospět) dosáhnout určité úrovně
mzda d. v tomto oboru 80 tisíc
d. pokročilého věku (NA)

ACT(.1) PAT(.2;aby[.v]) ?ORIG(na+6;od+2)
✓ (dobrat se, domoci se) dosáhl na něm slibu
dosáhli na sobě slibu (NA)

ACT(.1) DPHR(svůj-1.2)
✓ (docílit) dosáhl svého (NA)

ACT(.1) DIR3(*)
✓ (sáhnout, dotknout se) dosáhl na strop
rukou.MEANS (NA)

Figure 3.5: PDT-VALLEX in the TrEd editor

the users of PDT 1.0 we provide the data update that adds all the changed and new information to the original data. The update resides in the directory `data/pdt1.0-update`. The patch is restricted to the CSTS format only, old FS files cannot be updated.

The changes include:

- corrections of various errors on morphological and analytical layer,
- corrections of spelling errors,
- added human morphological annotation to all files.

Requirements for applying the patch. To update the data, you need two GNU tools, `gunzip` and `patch`. *On Linux*, these tools are usually already installed. *On Windows*, please download GNU `patch`[8] (other versions of it might not work). `gunzip` on Windows seems to work both from the Cygwin[9] distribution as well as from GNU[10]. PDT 2.0 CD-ROM contains a copy of Cygwin `gunzip.exe` in the directory `tools/tred/bin/`, so you can simply use that one.

Applying the patch to all data directories. PDT 1.0 CD-ROM contains several overlapping (by means of hard-links) subsets of the data in subdirectories of the directory *PDT_1.0_CD-ROM*/`Corpora/PDT_1.0/Data/`. All of them except for `fs/` and `fs-am/` have to be patched. *On Linux*, a script that patches all the subdirectories at once is available in `data/pdt1.0-update/linux-apply-patch.sh`. Run the script and follow the instructions. *On Windows*, we cannot provide a secure means to automatically apply the patch to all data directories. Please follow the instructions below and apply the patch manually to all the data subdirectories you need.

Applying the patch to a single data directory.

1. Copy files in a subdirectory of `Corpora/PDT_1.0/Data/` (except for `fs/` and `fs-am/`) on the PDT 1.0 CD-ROM to a new working directory.
2. Switch to this directory: **cd *`the_working_directory`***
3. Gunzip all the files: **gunzip *.gz**

[8] <http://gnuwin32.sourceforge.net/packages/patch.htm>
[9] <http://cygwin.com>
[10] <http://gnuwin32.sourceforge.net/packages/gzip.htm>

4. Apply the patch: **gunzip -c *PDT_2.0_CD-ROM*/data/pdt1.0-update/pdtpatch.gz** | **patch -p1 -t**

   The `-t` flag is required when patching incomplete directories, i.e. directories that do not contain all PDT 1.0 data files. This flag instructs `patch` to skip missing files without prompting the user. *On Windows,* make sure to add the flag `--binary` to the **patch** command. Otherwise, patching the files might fail.

Chapter 4

Tools

One of the two main purposes of PDT 2.0 (see the Section 1.1) is to give linguists a large number of real-world examples of (not only) the phenomena described previously by various theoretical works on the topic of dependency, tectogrammatical description and on the functional generative description approach in general. Without an intuitive search tool, however, such corpus would be of no or only a limited use.

Naturally, there are many ways how to do it. For instance, the most complex searches can be performed using `btred/ntred`, but programming skills (specifically, the knowledge of Perl and the `btred/ntred` interface) are necessary to do so. To most "ordinary" users we recommend *`Netgraph`*, a tool designed and developed for searching PDT 1.0 and PDT 2.0.

4.1 Searching trees: `Netgraph`

`Netgraph` is a client-server application that allows multiple users to search PDT 2.0 on-line and simultaneously through Internet. `Netgraph` is designed for making the search as easy and intuitive as possible and still keeping the strong power of the query language.

The `Netgraph` server and client communicate to each other through Internet. The server searches the treebank (the treebank and the server are located on the same computer or local network). The client serves as the front-end for users and may be located at any node in the Internet. It sends user queries to the server and receives results from it. Both the server and the client can, of course, also reside at the same computer.

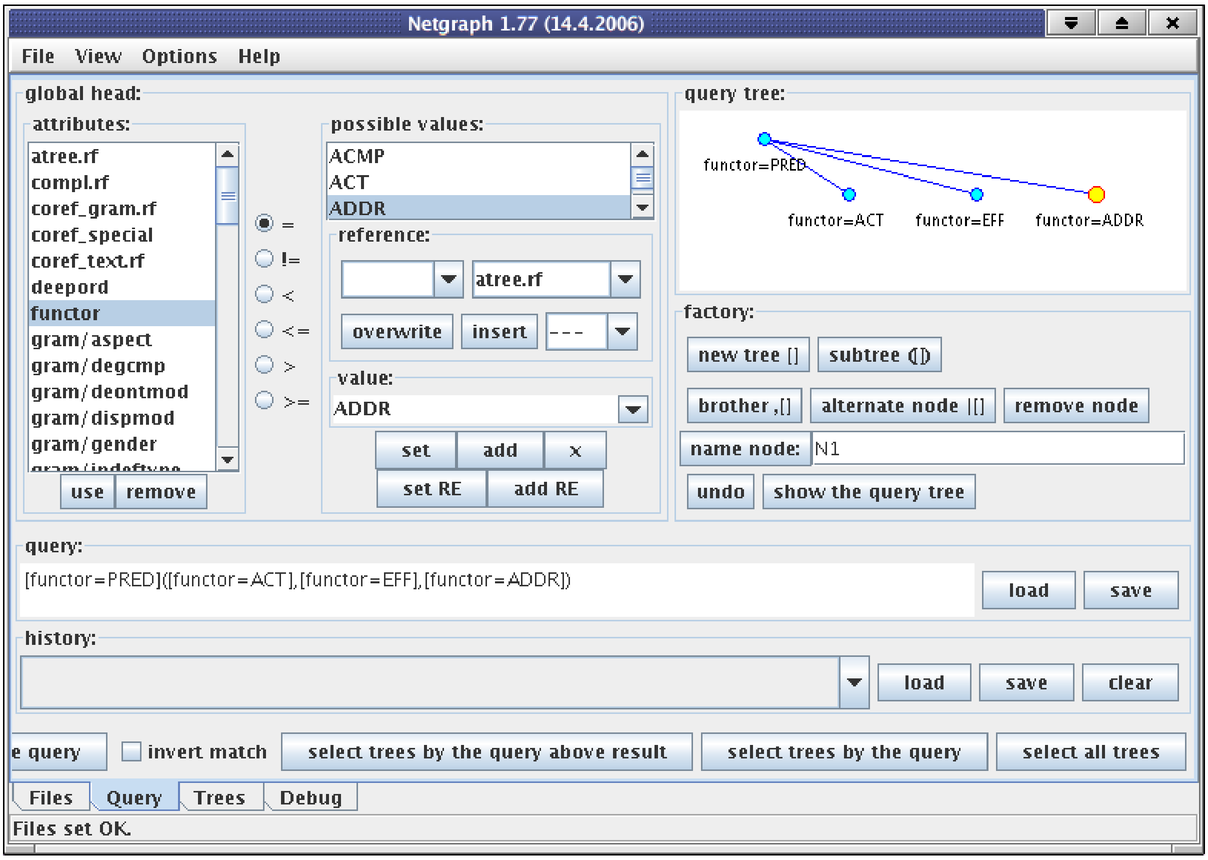
`Netgraph` server is written in C and C++ and works smoothly on Linux, other Unix-like systems, and Apple Mac OS. (An experimental version exists for MS Windows, too.) It requires the treebank in the FS format, encoded in UTF-8. `Netgraph` server allows setting user accounts with various access permissions.

`Netgraph` client is written in Java and is platform-independent. It exists in two forms—as a stand-alone Java application (which is full-featured and needs to be installed first, along with a Java 2 Runtime Environment), and as a Java applet (which provides the full search power but runs in a web browser without installation; it requires a Java 2 plug-in to be installed in your browser, though).

A query in `Netgraph` is a single node or a subtree with user-specified properties which s/he wishes to find in the corpus. Searching the corpus thus means searching for sentences (in the form of annotated trees, of course) that contain the query as a subtree. The properties of the subtree which the user can specify range from the most simple ones (such as searching for all trees in the corpus that contain a given word) to very elaborate ones (such as searching for all sentences with any verb that is modified by an Addressee which is not in the dative case *and* by at least one directional adverbial, etc.) This simple definition is extended using so called *meta-attributes* in order to allow setting even more complex queries. The meta-attributes allow setting conditions on transitive edges, optional nodes, position of the query nodes in the trees, size of trees, order of nodes, relations between attributes at different nodes in the trees, negation, and many other such things.

Queries in `Netgraph` are created using a user-friendly graphic environment. An example of such a query is in Figure 4.1. In this query, we are interested in all trees containing a node labeled as predicate and governing at least three nodes labeled as Actor, Effect, and Addressee. There is no condition on the order of the nodes.

One of the results (sent back by the server) may look like as in Figure 4.2.



Netgraph 1.77 (14.4.2006)

global head:

attributes: functor

● =

possible values: ADDR

reference: atree.rf

value: ADDR

query tree:

functor=PRED

functor=ACT functor=EFF functor=ADDR

name node: N1

query: [functor=PRED]([functor=ACT],[functor=EFF],[functor=ADDR])

history:

□ invert match

Files set OK.

Figure 4.1: Creating a query in Netgraph

The nodes matching the query are highlighted by yellow and green color. As you can see, the predicate in the result has got more sons than we have specified in the query. This is in accordance with the definition of searching in Netgraph—it is sufficient that the query tree is included in the result tree only as a subtree. Also note that the order of the nodes in the result is different from their order in the query. Meta-attributes allow controlling both the real number of sons and the order of nodes, if required.

For information about how to install Netgraph, see quick installation instructions for Netgraph client and quick installation instructions for Netgraph server. You should also read the Netgraph Client Manual and the Netgraph Server Installation Manual.

Please note that you need to install Netgraph server only if you want to search your own tree corpus. For searching PDT 2.0, a powerful Netgraph server is provided by Institute of Formal and Applied Linguistics[1] at `quest.ms.mff.cuni.cz` on port `2200`. It is accessible for *anonymous* user via Internet and you can connect to it using Netgraph client (see quick installation instructions for Netgraph client).

For more information about Netgraph, read the Netgraph client manual. If you want a full non-anonymous access to the server, or to get more information, updates and news please visit the Netgraph home page[2].

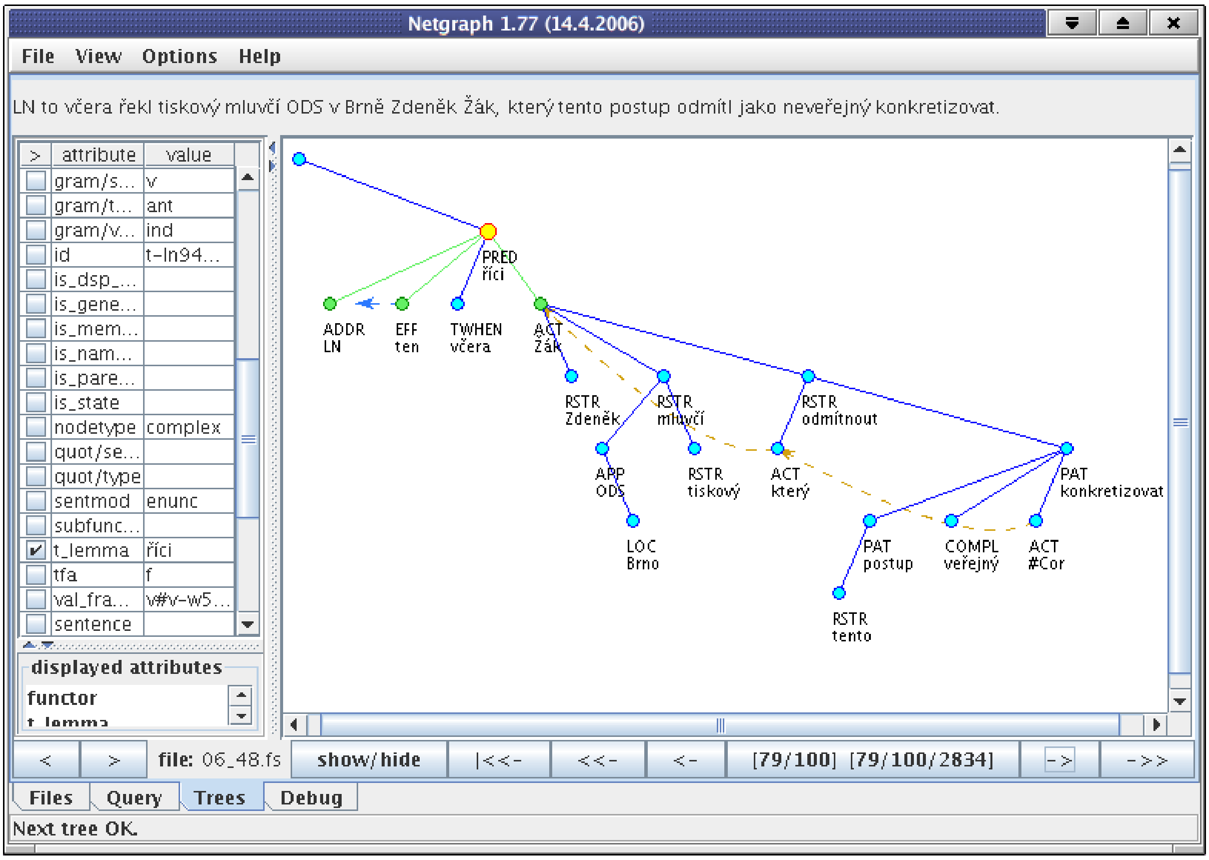
4.2 Viewing (browsing) trees: TrEd

The most perspicuous and most comfortable visualization of the data is provided by TrEd. It originally served as the main annotation tool, but it can be used as a data browser as well, with several types of search functions available. For TrEd installation instructions, see TrEd documentation.

To open files in TrEd select the **File** menu and click **Open**. Select any of the `*.t.gz` files (i.e. a file with tectogrammatical annotation of a document) and TrEd will open it and immediately display the tree for the first sentence in the file.

[1] <http://ufal.mff.cuni.cz>

[2] <http://quest.ms.mff.cuni.cz/netgraph>



Netgraph 1.77 (14.4.2006)

LN to včera řekl tiskový mluvčí ODS v Brně Zdeněk Žák, který tento postup odmítl jako neveřejný konkretizovat.

>	attribute	value
□	gram/s...	v
□	gram/t...	ant
□	gram/v...	ind
□	id	t-ln94...
□	is_dsp_...	
□	is_gene...	
□	is_mem...	
□	is_nam...	
□	is_pare...	
□	is_state	
□	nodetype	complex
□	quot/se...	
□	quot/type	
□	sentmod	enunc
□	subfunc...	
☑	t_lemma	říci
□	tfa	f
□	val_fra...	v#v-w5...
□	sentence	

displayed attributes: functor, t_lemma

file: 06_48.fs

[79/100] [79/100/2834]

Next tree OK.

Figure 4.2: A result tree in Netgraph

For an illustration of a typical screen, see Figure 4.3; the Czech sentence is *Kde jsou auta, tam je kšeft.* (lit.: *Where are cars, there is business.*).

1. Here you can see one or more windowing frames. Each frame displays one tree.

2. In this field you see the plain textual form of the sentence displayed in the currently selected frame.

3. Status line. It displays various information depending on the current context.

4. Current context. You can change the context by clicking on the name and selecting a different one from the list (such as PML_T_Edit).

5. Current stylesheet. It can be changed in a similar way as the context.

6. Click here to edit the stylesheet.

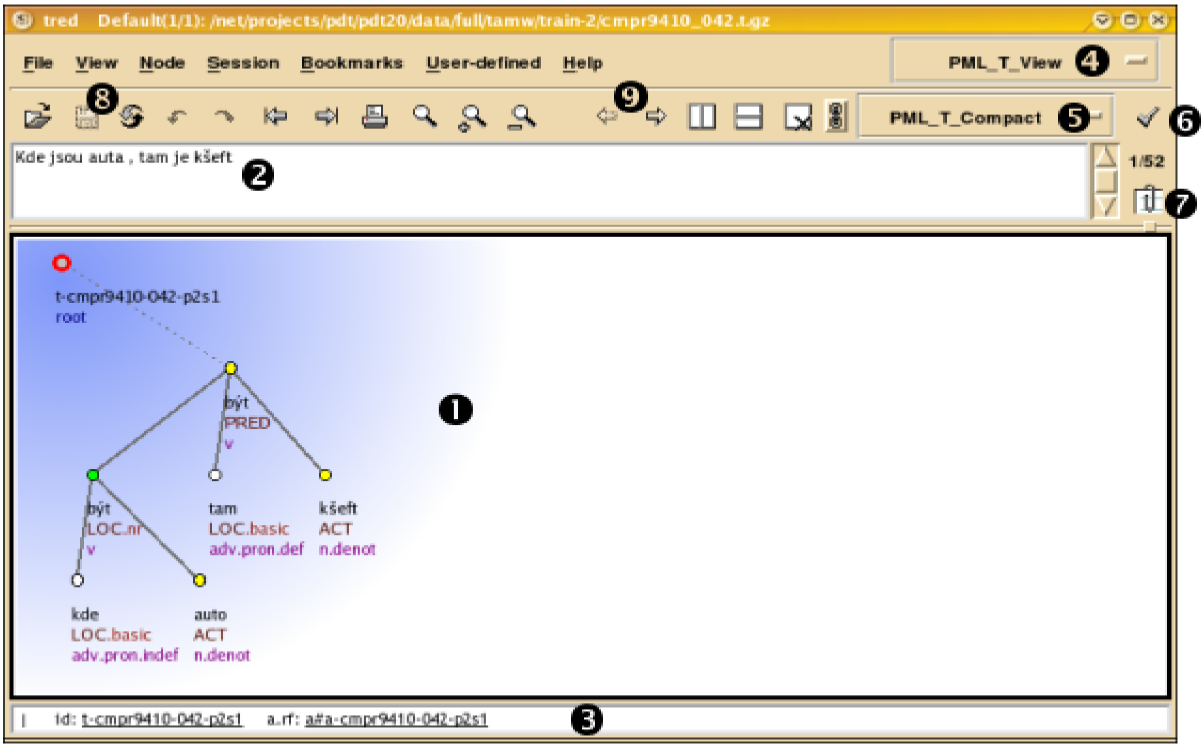
7. Click here to get the list of all the sentences in the current file. The index of the current tree in the current file is displayed above this button.

8. Buttons to open, save and reload a file. The icons mean *Undo*, *Redo*, *Previous* and *Next File*, *Print*, *Find*, *Find Next*, *Find Previous*.

9. Buttons for moving to the previous/next tree in the current file and for frame management.

By default, PDT 2.0 tectogrammatical files in the PML format are opened in the PML_T_View context that does not allow the user to edit anything. If you want to edit the files, you can switch to the PML_T_Edit context. In both contexts, two style-sheets are provided. The default one is PML_T_Compact but you can use the PML_T_Full if you want to see more details. For information on contexts and stylesheets, see the documentation of TrEd macros.

In any context, select **View** → **List of Named Macros** from the menu to see the list of all macros defined in the context and their possible keyboard shortcuts.



tred Default(1/1): /net/projects/pdt/pdt20/data/full/tamw/train-2/cmpr9410_042.t.gz

Kde jsou auta , tam je kšeft

id: t-cmpr9410-042-p2s1 a.rf: a#a-cmpr9410-042-p2s1

Figure 4.3: Tectogrammatical tree in TrEd

4.3 Automatic tree processing: btred/ntred

Whereas Netgraph (Section 4.1) allows a non-programmer to comfortably search the PDT trees, and the tree editor TrEd (Section 4.2) allows for a quick, comfortable and customizable browsing and viewing of linguistic tree structures, for tool developers and programmers in general full access to the data becomes necessary. You can always process the data directly (it is XML, after all), but we recommend you to access the data via btred/ntred Perl-based interface tailored for the PDT 2.0 data. *btred* is a Perl program that allows the user to apply another Perl program (called *btred macro*) on the data stored in one of the PDT formats. *ntred* is a client-server version of btred and it is suited for data processing on multiple machines in parallel (mnemonics for btred/ntred: “b” stands for “batch processing”, “n” stands for “networked processing”).

If you follow the above recommendation, you get several advantages:

- Object-oriented tree representation, which is used in btred/ntred environment, offers a rich repertory of basic functions for tree traversing and for many other basic operations on trees; besides that, several highly non-trivial functions are provided, suitable for linguistically motivated traversing of trees (reflecting e.g. the interplay of dependency and coordination relations).

- btred/ntred technology was extensively used by several programmers during the development of PDT 2.0; this long-time experience has led to many improvements and makes the tools and accompanying libraries reasonably stable.

- If you have more computers at your disposal, you can use ntred and process the data in parallel, which makes the computation considerably faster. Depending on the situation, it may shorten the time needed for passing the whole PDT 2.0 to just a few seconds (with only about 10 CPUs available for the distributed btred to run on).

- Programmers can use btred/ntred (in combination with TrEd) as a powerful and fast search engine—you write a macro which finds the treebank positions you are interested in, run it by ntred and then simply view the retrieved positions in TrEd.

- All you need to become fluent in writing btred/ntred macros is to know the basics of Perl syntax and to remember a few names of variables and subroutines predefined in the btred/ntred environment.

- Once you learn the `btred/ntred` style of work, you can re-use all of its benefits when processing data from other treebanks (be they dependency- or constituency-based).

Read the `btred/ntred` tutorial to get started. See also the `btred` and `ntred` manual pages.

4.4 Converting data between formats

4.4.1 Conversion between the PDT formats

Conversion between data formats is a tough task unless all the formats can bear exactly the same amount of information. Unfortunately, this is not the case of data formats that emerged over the years of history of PDT. Thus, various tools are provided to make at least some of the conversions easier. They may also serve as examples for more complex transformations required for a particular purpose. For a full description, see PDT 2.0: internal format conversion tools.

In the distribution, the scripts are located in the directory `tools/format-conversions/pdt_formats`. Most of the scripts also require the `btred` tool from the `TrEd` toolkit.

The following types of conversions are supported:

- conversion of PDT1.0-like analytical annotation to PML,
- conversion of a PML a-data instance to CSTS,
- conversion of a PML m-data instance to CSTS,
- conversion of PDT 2.0 data to FS for Netgraph,
- conversion of PDT 2.0 data to a binary Perl Storable format (for speed).

4.4.2 Conversion from formats of other treebanks

The conversion scripts are provided for the purposes of importing Penn Treebank and Negra corpus formats to FS format. The conversion scripts reside in the directory `tools/format-conversions/from_negra+ptb`. You can read their short documentation.

Please note that *no conversion of annotation schemata* is performed. In other words, constituency trees remain constituency trees, no dependency structure is automatically derived.

4.5 Parsing Czech: from plain text to PDT-formatted dependency trees

Together with the data, we also supply tools which perform automatic annotation—they create dependency trees represented at the analytical layer from unannotated Czech sentences. The tools are stored in the directory `tools/machine-annotation`. The tools perform the following tasks in a sequence:

- tokenization of the input plain text and segmentation into sentences,
- morphological analysis and tagging (morphological disambiguation),
- dependency parsing,
- analytical (dependency) function assignment for all nodes of the parsed tree.

There are no tools (yet) for the continuation of this process to the tectogrammatical layer. Please watch the web page <`http://ufal.ms.mff.cuni.cz/pdt2.0update/`> with PDT 2.0 updates for updates and new tools.

You can read the detailed description of the tools.

4.6 Creating data for parser development

When developing a new parser, it is important to evaluate its performance not only on the human-annotated m-layer files, but also on the machine-annotated ones. Due to space reasons it was not possible to include machine-annotated m-layer files in the CD-ROM. Instead, a special tool for generating the data for parser development and evaluation is provided.

The tool resides in the directory `tools/machine-annotation/for_parser_devel/`. It is run by the command

```
run_for_parser_devel input_directory output_directory
```

The input directory has to have the same structure like `data/full/`, which typically be its first argument. The tool copies the whole directory structure of the input directory into the output directory. It also copies all the data files except for the m-layer ones, which are substituted with the newly generated m-layer files. The new m-files contain automatically assigned lemmas and tags. However, note that the new files are not equivalent to what would be obtained by machine annotation applied directly on a plain text, since the new files preserve the sentence and token boundaries as well as identifiers of m-layer units as contained in the manually annotated files.

4.7 Macros for error detection

Although the annotators have seen every node of every tree (often more than once), they have still made some errors. Some of them have been caused by an inadvertence of the annotators, other errors surfaced because the annotation rules evolved and changed during the annotation process, while the data were not re-annotated every time a rule changed. Therefore, a large set of programs (`TrEd/btred/ntred` macros, see Section 4.2) was developed during the annotation phase and during the checking phase, each macro searching the data for a violation of a rule, an invariant or a suspicious annotation, reporting the affected corpus positions. The data have then been manually or automatically repaired or the macro was changed if necessary.

Note: For help on writing macros for `TrEd`, see the documentation of `TrEd`.

The macros were divided into three groups: find, fix and check. Macros from the *find* group were just searching for all the suspicious data. Macros from the *fix* group were used where automatic reparation was possible, such as when an unambiguous, defined annotation rule change appeared in the middle of the annotation process. The last group (*check*) contained macros similar to those in the *find* group, but they included lists of exceptions to the general rules. (In fact, there was also another group called *misc* that contained miscellaneous macros and scripts.)

The macros were also divided into groups with respect to which layer of annotation they apply to (see Chapter 2, “Layers of annotation” for details on layers).

The macros from the *check* group are included in directory `tools/checks`. **Warning:** These macros are not intended to be used anymore because the format of the data has changed but one can browse them to get the taste of what kind of checks have been applied to PDT 2.0, and what macros can be programmed to deal with tree structures.

Chapter 5

Documentation

Since links to documentation of the tools, data formats etc. are scattered throughout the whole PDT guide, we have collected them here in a slightly more structured manner.

- PDT guide (this is what you are reading)
  - in English
    - HTML version: `doc/pdt-guide/en/html/index.html`
    - PDF version: `doc/pdt-guide/en/pdf/pdt-guide.pdf`
  - in Czech
    - HTML version: `doc/pdt-guide/cz/html/index.html`
    - PDF version: `doc/pdt-guide/cz/pdf/pdt-guide.pdf`
- Annotation Manuals (see also Chapter 2, “Layers of annotation”)
  - Manual for Morphological Annotation
    - in English
      - HTML version: `doc/manuals/en/m-layer/html/index.html`
      - PDF version: `doc/manuals/en/m-layer/pdf/m-man-en.pdf`
  - Manual for Analytical Annotation
    - in English
      - HTML version: `doc/manuals/en/a-layer/html/index.html`
      - PDF version: `doc/manuals/en/a-layer/pdf/a-man-en.pdf`
    - in Czech
      - HTML version: `doc/manuals/cz/a-layer/html/index.html`
      - PDF version: `doc/manuals/cz/a-layer/pdf/a-man-cz.pdf`
  - Manual for Tectogrammatical Annotation
    - in English
      - HTML version: `doc/manuals/en/t-layer/html/index.html`
      - PDF version: `doc/manuals/en/t-layer/pdf/t-man-en.pdf`
    - in Czech
      - HTML version: `doc/manuals/cz/t-layer/html/index.html`
      - PDF version: `doc/manuals/cz/t-layer/pdf/t-man-cz.pdf`
- Data (see also Section 3.4)
  - CSTS
    - complete description: `doc/data-formats/csts/html/DTD-HOME.html`

        - DTD: `doc/data-formats/csts/csts.dtd`
    - FS, format specification: `doc/data-formats/fs/index.html`
    - PML
        - complete description:
            - HTML version: `doc/data-formats/pml/index.html`
            - PDF version: `doc/data-formats/pml/pml_doc.pdf`
        - schemata: `data/schemas`
    - PML markup (including description of node attributes): `doc/data-formats/pml-markup/index.html`
    - PDT-VALLEX, physical structure: `doc/data-formats/pdt-vallex/pdt-vallex-struct.html`

- Tools (see also Chapter 4, "Tools")
    - `TrEd`, `btred/ntred`
        - `TrEd`, manual: `doc/tools/tred/index.html`
        - `btred`, man pages: `doc/tools/tred/btred.html`
        - `ntred`, man pages: `doc/tools/tred/ntred.html`
        - `btred/ntred` tutorial: `doc/tools/tred/bn-tutorial.html`
        - `TrEd` macros: `doc/tools/tred/PML_mak.html`
    - `Netgraph:`
        - `Netgraph` Client Quick Installation: `doc/tools/netgraph/README_QUICK_INSTALL_CLIENT`
        - `Netgraph` Client Manual: `doc/tools/netgraph/netgraph_manual.html`
        - `Netgraph` Server Quick Installation: `doc/tools/netgraph/README_QUICK_INSTALL_SERVER`
        - `Netgraph>` Server Installation Manual: `doc/tools/netgraph/netgraph_server_install.html`
    - Conversion scripts:
        - From Penn Treebank and Negra formats: `doc/tools/format-conversions/from_negra+ptb/readme.txt`
        - Between the PDT formats: `doc/tools/format-conversions/pdt_formats/index.html`
    - Machine annotation (tokenization, morphology, parsing): `doc/tools/machine-annotation/index.html`

- Publications (see also Chapter 6, "Publications")
    - BibTeX items: `publications/pdt.bib`

Chapter 6

Publications

The list contains publications documenting:

- research done mainly before the PDT project had started and which was crucial for the formulation of the annotation strategy (Section 6.1),
- a complete list of papers on building PDT 2.0 (Section 6.2),
- tools, such as annotation editors, search systems, and natural language processing procedures (Section 6.3).

The general publications in Section 6.2 are sorted by the year when they appeared so that one can get a chronological picture of the work on PDT how the time went on. In other sections, publications are listed in the typical way, i.e. in an alphabetical order based on the last name of the first author.

Most of the publications are available in an electronic form (both PDF and Postscript files) as indicated by every publication. The electronic versions are author's copies that are provided on personal requests and thus they are designated *for personal use only*. BibTeX items of all the publication listed here are also available.

6.1 Theoretical background of PDT

- Eva Hajičová: *Issues of Sentence Structure and Discourse Patterns.* Charles University, Prague, Czech Republic, 1993. **Available:** BibTeX

- Eva Hajičová, Jarmila Panevová: "Valency (case) frames." In: P. Sgall (ed.): *Contributions to Functional Syntax, Semantics and Language Comprehension*, Prague:Academia, 1984, pp. 147–188. **Available:** BibTeX

- Eva Hajičová, Barbara H. Partee, Petr Sgall: *Topic-focus articulation, tripartite structures, and semantic content.* Amsterdam:Kluwer, 1998. **Available:** BibTeX

- Jarmila Panevová: "On verbal frames in Functional generative description I." In: *Prague Bulletin of Mathematical Linguistics*, 22, MFF UK, Prague, Czech Republic, 1974, pp. 3–40. **Available:** PDF, PS, BibTeX

- Jarmila Panevová: "On verbal frames in functional generative description II." In: *Prague Bulletin of Mathematical Linguistics*, 23, MFF UK, Prague, Czech Republic, 1975, pp. 17–52. **Available:** PDF, PS, BibTeX

- Jarmila Panevová: *Formy a funkce ve stavbě české věty.* Prague:Academia, 1980. **Available:** BibTeX

- Vladimír Petkevič: "A new dependency based specification of underlying representations of sentences." In: *Theoretical Linguistics*, 14, 1987, pp. 143–172. **Available:** BibTeX

- Vladimir Petkevič: "A New Formal Specification of Underlying Representations." In: *Theoretical Linguistics*, 21, 1995, pp. 7–61. **Available:** BibTeX

- Petr Sgall: *Generativní popis jazyka a česká deklinace.* Prague:Academia, 1967. **Available:** BibTeX

- Petr Sgall: *Contributions to Functional Syntax, Semantics and Language Comprehension.* Prague:Academia, 1984. **Available:** BibTeX

- Petr Sgall: "Underlying Structure of Sentence and its Relation to Semantics." In: T. Reuther (ed.): *Wiener Slawistischer Almanach. Sonderband 33*, 1992, pp. 273–282. **Available:** BibTeX

- Petr Sgall: "Valency and Underlying Structure. An alternative view on dependency." In: L. Wanner (ed.): *Recent Trends in meaning-text theory*, Amsterdam/Philadelphia: Benjamins, 1997, pp. 149–166. **Available:** BibTeX

- Petr Sgall, Eva Hajičová, Jarmila Panevová: *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects.* Dordrecht:Reidel Publishing Company and Prague:Academia, 1986. **Available:** BibTeX

- Vladimír Šmilauer: *Novočeská skladba.* Státní pedagogické nakladatelství, Prague, Czech Republic, 1969. **Available:** BibTeX

6.2 PDT 2.0

6.2.1 General information

Motivation to build PDT

- Jan Hajič, Eva Hajičová, Alexander Rosen: "Formal Representation of Language Structures." In: *TELRI Newsletter*, 3, 1996, pp. 12–19. **Available:** PDF, PS, BibTeX

2000

- Jan Hajič, Alena Böhmová, Eva Hajičová, Barbora Vidová Hladká: "The Prague Dependency Treebank: A Three-Level Annotation Scenario." In: A. Abeillé (ed.): Treebanks: *Building and Using Parsed Corpora*, Amsterdam:Kluwer, 2000, pp. 103–127. **Available:** PDF, PS, BibTeX

- Jarmila Panevová: *Building an electronic language database nowadays: The Prague Dependency Treebank. 2000. **Available:** PDF, PS, BibTeX*

2001

- Jan Hajič, Eva Hajičová, Petr Pajas, Jarmila Panevová, Petr Sgall, Barbora Vidová Hladká: "Prague Dependency Treebank 1.0 (Final Production Label)." In: *CD-ROM*, CAT: LDC2001T10, ISBN 1-58563-212-0, Linguistic Data Consortium, 2001. **Available:** BibTeX

- Jan Hajič, Petr Pajas, Barbora Vidová Hladká: "The Prague Dependency Treebank: Annotation Structure and Support." In: *Proceedings of the IRCS Workshop on Linguistic Databases*, University of Pennsylvania, Philadelphia, USA, 2001, pp. 105–114. **Available:** PDF, PS, BibTeX

- Eva Hajičová, Jan Hajič, Martin Holub, Petr Pajas, Veronika Kolářová-Řezníčková, Petr Sgall, Barbora Vidová Hladká: "The Current Status of the Prague Dependency Treebank." In: V. Matoušek, P. Mautner, R. Mouček, K. Taušer (eds.): *Proceedings of the 5th International Conference on Text, Speech and Dialogue, Železná Ruda–Špičák, Czech Republic*, Springer-Verlag Berlin Heidelberg New York, 2001, pp. 11–20. **Available:** PDF, PS, BibTeX

- Eva Hajičová, Petr Sgall: "A reusable corpus needs syntactic annotations: Prague Dependency Treebank." In: *A rainbow of corpora–corpus linguistics and the languages of the world*, Munich: Licom-Europa, 2001, pp. 37–48. **Available:** PDF, PS, BibTeX

2002

- Eva Hajičová: "Theoretical description of language as a basis of corpus annotation: The case of Prague Dependency Treebank." In: E. Hajičová, P. Sgall, J. Hana, T. Hoskovec (eds.): *Prague Linguistic Circle Papers*, (4), Amsterdam/Philadelphia:John Benjamins, 2002, pp. 111–127. **Available:** BibTeX

- Jarmila Panevová, Eva Hajičová, Petr Sgall: "K nové úrovni bohemistické práce: Využití anotovaného korpusu. Část 1." In: *Slovo a slovesnost*, 63, Czech Academy of Science, Prague, 2002, pp. 161–177. **Available:** PDF, PS, BibTeX

- Jarmila Panevová, Eva Hajičová, Petr Sgall: "K nové úrovni bohemistické práce: Využití anotovaného korpusu. Část 2." In: *Slovo a slovesnost*, 63, Czech Academy of Science, Prague, 2002, pp. 241–262. **Available:** PDF, PS, BibTeX

- Barbora Vidová Hladká: "Pražský závislostní korpus aneb Co tady před padesáti lety nebylo." In: *Pokroky matematiky, fyziky a astronomie*, 47, (4), Jednota českých matematiků a fyziků, 2002, pp. 298–306. **Available:** PDF, PS, BibTeX

2003

- Alena Böhmová, Eva Hajičová: "Large Language Data and the Degrees of Automation." In: E. Hajičová, A. Kotěšovcová, J. Mírovský (eds.): *Proceedings of XVII International Congress of Linguists, CD-ROM*, Matfyzpress, MFF UK, Prague, Czech Republic, 2003. **Available:** PDF, PS, BibTeX

2004

- Jan Hajič: *Complex Corpus Annotation: The Prague Dependency Treebank.* Jazykovedný ústav Ľ. Štúra, SAV, Bratislava, Slovakia, 2004. **Available:** PDF, PS, BibTeX

2005

- Petr Pajas, Jan Štěpánek: "A Generic XML-Based Format for Structured Linguistic Annotation and Its Application to Prague Dependency Treebank 2.0." In: *UFAL Technical Report*, 29, MFF UK, Prague, Czech Republic, 2005. **Available:** PDF, PS, BibTeX

6.2.2 Morphological layer

- Jan Hajič: *Disambiguation of Rich Inflection (Computational Morphology of Czech).* Karolinum, Charles University Press, Prague, Czech Republic, 2004. **Available:** BibTeX

- Dan Zeman, Jiří Hana, Hana Hanová, Jan Hajič, Emil Jeřábek, Barbora Vidová Hladká: "A Manual for Morphological Annotation, 2nd edition (html)." In: *ÚFAL Technical Report*, 27, MFF UK, Prague, Czech Republic, 2005. **Available:** PDF, PS, BibTeX

6.2.3 Analytical layer

- Jan Hajič: "Building a Syntactically Annotated Corpus: The Prague Dependency Treebank." In: E. Hajičová (ed.): *Issues of Valency and Meaning. Studies in Honour of Jarmila Panevová*, Karolinum, Charles University Press, Prague, Czech Republic, 1998, pp. 106–132. **Available:** PDF, PS, BibTeX

- Jan Hajič, Eva Hajičová: "Syntactic tagging in the Prague Dependency Treebank." In: R. Marcinkeviciene, N. Volz (eds.): *Proceedings of the Second European Seminar "Language Applications for a Multilingual Europe"*, TELRI, Kaunas, Lithuania, 1997, pp. 55–68. **Available:** PDF, PS, BibTeX

- Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall: "Syntax v Českém národním korpusu." In: *Slovo a slovesnost*, Czech Academy of Science, Prague, 1998, pp. 168–177. **Available:** BibTeX

- Jan Hajič, Jarmila Panevová, Eva Buráňová, Zdeňka Urešová, Alla Bémová: *Anotace Pražského závislostního korpusu na analytické rovině: pokyny pro anotátory (html). 1999. **Available:** PDF, PS, BibTeX*

- Eva Hajičová, Zdeněk Kirschner, Petr Sgall: "A Manual for Analytical Layer Annotation of the Prague Dependency Treebank (English translation) (html). 1999. **Available:** PDF, PS, BibTeX "

- Roman Ondruška, Jarmila Panevová, Jan Štěpánek: "An Exploitation of the Prague Dependency Treebank: A Valency Case." In: K. Simov, P. Osenova (eds.): *Proceedings of the Workshop on Shallow Processing of Large Corpora*, UCREL, Lancaster University, Lancaster, Great Britain, 2003, pp. 69–77. **Available:** PDF, PS, BibTeX

6.2.4 Tectogrammatical layer

Annotation structure

- Alena Böhmová: "Automatic Procedures in Tectogrammatical Tagging." In: *Prague Bulletin of Mathematical Linguistics*, 76, MFF UK, Prague, Czech Republic, 2001, pp. 23–34. **Available:** PDF, PS, BibTeX

- Alena Böhmová, Silvie Cinková, Eva Hajičová: *A Manual for Tectogrammatical Layer Annotation of the Prague Dependency Treebank (English translation) (html). 2005.* ***Available:*** *PDF, PS, BibTeX*

- Alena Böhmová, Petr Sgall: "Automatic procedures in tectogrammatical tagging." In: *Proceedings of the Workshop on Linguistically Interpreted Corpora, 18th International Conference on Computational Linguistics, Saarbrücken, Germany*, 2000, pp. 65–70. **Available:** PDF, PS, BibTeX

- Eva Hajičová: "Prague Dependency Treebank: From analytic to tectogrammatical annotations." In: P. Sojka, V. Matoušek, K. Pala, I. Kopeček (eds.): *Proceedings of the 2nd International Conference on Text, Speech and Dialogue, Brno, Czech Republic*, Springer-Verlag Berlin Heidelberg New York, 1998, pp. 45–50. **Available:** PDF, PS, BibTeX

- Eva Hajičová, Jiří Havelka, Petr Sgall, Kateřina Veselá, Daniel Zeman: "Issues of Projectivity in the Prague Dependency Treebank." In: *Prague Bulletin of Mathematical Linguistics*, 81, MFF UK, Prague, Czech Republic, Prague, 2004, pp. 5–22. **Available:** PDF, PS, BibTeX

- Eva Hajičová, Petr Pajas: "Evaluation of Tectogrammatical Annotation of PDT." In: P. Sojka, I. Kopeček, K. Pala (eds.): *Proceedings of the 3rd International Conference on Text, Speech and Dialogue, Brno, Czech Republic*, Springer-Verlag Berlin Heidelberg New York, 2000, pp. 75–80. **Available:** BibTeX

- Eva Hajičová, Petr Pajas, Kateřina Veselá: "Corpus Annotation on the Tectogrammatical Layer: Summarizing the First Stages of Evaluations." In: *Prague Bulletin of Mathematical Linguistics*, 77, MFF UK, Prague, Czech Republic, Prague, 2002, pp. 5–18. **Available:** PDF, PS, BibTeX

- Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolářová-Řezníčková, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Magda Razímová, Petr Sgall, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, Zdeněk Žabokrtský: *Anotace Pražského závislostního korpusu na tektogramatické rovině: pokyny pro anotátory (html). 2005.* ***Available:*** *PDF, PS, BibTeX*

- Jarmila Panevová, Alena Böhmová, Petr Sgall: "Syntactic Tagging: Procedure for the Transition from the Analytic to the Tectogrammatical Tree Structure." In: V. Matoušek, P. Mautner, J. Ocelíková, P. Sojka (eds.): *Proceedings of the 2nd International Conference on Text, Speech and Dialogue, Plzeň, Czech Republic*, Springer-Verlag Berlin Heidelberg New York, 1999, pp. 34–38. **Available:** PDF, PS, BibTeX

- Jarmila Panevová, Eva Hajičová, Petr Sgall: "Tectogrammatics in corpus tagging." In: I. Kenesei, R. M. Harnish (eds.): *Perspectives on Semantics, Pragmatics, and Discourse; A Festschrift for Ferenc Kiefer (Pragmatics and Beyond new Series)*, (90), Amsterdam/Philadelphia:John Benjamins, 2001, pp. 294–299. **Available:** PDF, PS, BibTeX

- Jarmila Panevová, Veronika Kolářová-Řezníčková, Zdeňka Urešová: "The Theory of Control Applied to the Prague Dependency Treebank (PDT)." In: R. Frank (ed.): *Proceedings of the 6th International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+6)*, Universita di Venezia, Venezia, Italy, 2002, pp. 175–180. **Available:** PDF, PS, BibTeX

- Veronika Kolářová-Řezníčková: "PDT: Two Steps in Tectogrammatical Annotation with respect to some Issues of Deletion." In: *Prague Bulletin of Mathematical Linguistics*, 78, MFF UK, Prague, Czech Republic, Prague, 2002, pp. 37–52. **Available:** PDF, PS, BibTeX

- Petr Sgall, Jarmila Panevová, Eva Hajičová: "Deep Syntactic Annotation: Tectogrammatical Representation and Beyond." In: A. Meyers (ed.): Proceedings of the HLT-NAACL 2004 Workshop: *Frontiers in Corpus Annotation*, Association for Computational Linguistics, Boston, Massachusetts, USA, 2004, pp. 32–38. **Available:** PDF, PS, BibTeX

Topic-focus articulation

- Eva Hajičová: "The Prague Dependency Treebank: Crossing the Sentence Boundary." In: V. Matoušek, P. Mautner, J. Ocelíková, P. Sojka (eds.): *Proceedings of the 2nd International Conference on Text, Speech and Dialogue, Plzeň, Czech Republic*, Springer-Verlag Berlin Heidelberg New York, 1999, pp. 20–27. **Available:** PDF, PS, BibTeX
- Eva Hajičová: "Dependency-Based Underlying-Structure Tagging of a Very Large Czech Corpus." In: S. Kahane (ed.): *Special issue of TAL journal, Grammaires de Dépendence / Dependency Grammars*, Paris:Hermes, 2000, pp. 57–78. **Available:** PDF, PS, BibTeX
- Eva Hajičová, Petr Sgall: *Degrees of Contrast and the Topic-Focus Articulation. (1),* Berlin:Walter de Gruyter, 2004, pp. 1–13. **Available:** PDF, PS, BibTeX
- Eva Hajičová, Petr Sgall, Eva Buráňová: "Topic-Focus Articulation and degrees of salience in the Prague Dependency Treebank." In: A. Carnie, H. Harley, M. Willie (eds.): *Formal Approaches to Function in Grammar. In honor of Eloise Jelinek, Arizona*, Amsterdam/Philadelphia:John Benjamins, Amsterdam/Philadelphia, 2003, pp. 165–177. **Available:** PDF, PS, BibTeX
- Eva Hajičová, Petr Sgall, Eva Buráňová: "Tagging of very large corpora: Topic-Focus Articulation." In: *Proceedings of the 18th International Conference on Computational Linguistics, Saarbrücken, Germany*, 2000, pp. 139–144. **Available:** PDF, PS, BibTeX
- Lucie Kučová, Eva Hajičová, Kateřina Veselá, Jiří Havelka: "Topic-focus articulation and anaphoric relations: A corpus based probe." In: (ed.): *Prague Bulletin of Mathematical Linguistics*, 84, MFF UK, Prague, Czech Republic, 2005, pp. 5–12. **Available:** PDF, PS, BibTeX
- Petr Sgall: "Topic-Focus Articulation in Corpus Annotation." In: W. Menzel, C. Vertan (eds.): *Natural language processing between linguistic inquiry and system engineering,* Editura Universitatii Alexandru Ioan Cuza, Iasi, 2003, pp. 95–101. **Available:** PDF, PS, BibTeX
- Kateřina Veselá, Jiří Havelka: "Anotování aktuálního členění věty v Pražském závislostním korpusu." In: *ÚFAL Technical Report*, 20, MFF UK, Prague, Czech Republic, 2003. **Available:** PDF, PS, BibTeX
- Kateřina Veselá, Jiří Havelka, Eva Hajičová: "Annotators' Agreement: The Case of Topic-Focus Articulation." In: *Proceedings of the 4th International Conference on Language Resources and Evaluation*, European Language Resources Association, Lisboa, Portugal, 2004, pp. 2191–2194. **Available:** PDF, PS, BibTeX

Coreference

- Lucie Kučová, Eva Hajičová: "Coreferential Relations in the Prague Dependency Treebank." In: (ed.): *Proceedings of the 5th International Conference on Discourse Anaphora and Anaphor Resolution 2004,*

  San Miguel, Azores, Sept. 23-24, 2004, 2005, pp. 94–102. **Available:** PDF, PS, BibTeX
- Lucie Kučová, Veronika Kolářová-Řezníčková, Zdeněk Žabokrtský, Petr Pajas, Oliver Čulo: "Anotování koreference v Pražském závislostním korpusu." In: *ÚFAL Technical Report*, 19, MFF UK, Prague, Czech Republic, 2003. **Available:** PDF, PS, BibTeX
- Jarmila Panevová, Eva Hajičová, Petr Sgall: "Coreference in Annotating a Large Corpus." In: M. Gavrilidou, G. Carayannis, S. Markantonatou, S. Piperidis, G. Stainhaouer (eds.): *Proceedings of the 2nd International Conference on Language Resources*, (I), European Language Resources Association, Athens, Greece, 2000, pp. 497–500. **Available:** PDF, PS, BibTeX

PDT-VALLEX

- Silvie Cinková, Veronika Kolářová-Řezníčková: "Nouns as Components of Support Verb Constructions in the Prague Dependency Treebank." In: *Korpusy a korpusová lingvistika v zahraničí a na Slovensku*, 2004. **Available:** PDF, PS, BibTeX
- Jan Hajič, Václav Honetschläger: "Annotation Lexicons: Using the Valency Lexicon for Tectogrammatical Annotation." In: *Prague Bulletin of Mathematical Linguistics*, 79–80, MFF UK, Prague, Czech Republic, 2003, pp. 61–86. **Available:** PDF, PS, BibTeX

- Jan Hajič, Jarmila Panevová, Zdeňka Urešová, Alevtina Bémová, Veronika Kolářová-Řezníčková, Petr Pajas: "PDT-VALLEX: Creating a Large-coverage Valency Lexicon for Treebank Annotation." In: J. Nivre, E. Hinrichs (eds.): *Proceedings of The Second Workshop on Treebanks and Linguistic Theories*, Vaxjo University Press, Vaxjo, Sweden, 2003, pp. 57–68. **Available:** PDF, PS, BibTeX

- Jan Hajič, Zdeňka Urešová: "Linguistic Annotation: from Links to Cross-Layer Lexicons." In: J. Nivre, E. Hinrichs (eds.): *Proceedings of The Second Workshop on Treebanks and Linguistic Theories*, Vaxjo University Press, Vaxjo, Sweden, 2003, pp. 69–80. **Available:** PDF, PS, BibTeX

- Václav Honetschläger: "Using a Czech Valency Lexicon for Annotation Support." In: V. Matoušek, P. Mautner (eds.): *Proceedings of the 6th International Conference on Text, Speech and Dialogue, České Budějovice, Czech Republic*, Springer-Verlag Berlin Heidelberg New York, 2003, pp. 120–126. **Available:** PDF, PS, BibTeX

- Markéta Lopatková, Jarmila Panevová: "Recent developments of the theory of valency in the light of the Prague Dependency Treebank." In: Mária Šimková (ed.): ,

  Veda Bratislava, Slovakia,

  2005. **Available:** PDF, PS, BibTeX

- Zdeňka Urešová: *The Verbal Valency in the Prague Dependency Treebank from the Annotator's Point of View.* Jazykovedný ústav Ľ. Štúra, SAV, Bratislava, Slovakia, 2004. **Available:** PDF, PS, BibTeX

- Zdeněk Žabokrtský: *Valency Lexicon of Czech Verbs (PhD thesis).*

  UFAL MFF UK, Prague, Czech Republic, 2005. **Available:** PDF, PS, BibTeX

6.3 Tools

6.3.1 Netgraph

- Jiří Mírovský, Roman Ondruška: "NetGraph System: Searching through the Prague Dependency Treebank." In: *Prague Bulletin of Mathematical Linguistics*, 77, MFF UK, Prague, Czech Republic, Prague, 2002, pp. 101–104. **Available:** PDF, PS, BibTeX

- Roman Ondruška, Jiří Mírovský, Daniel Průša: "Searching through Prague Dependency Treebank-Conception and Architecture." In: *Proceedings of The First Workshop on Treebanks and Linguistic Theories*, LML, Bulgarian Academy of Sciences and SfS, Tuebingen University, Sofia, Bulgaria and Tuebingen, Germany, 2002, pp. 114–122. **Available:** PDF, PS, BibTeX

6.3.2 Morphological analysis and tagging

- Jan Hajič: *Disambiguation of Rich Inflection (Computational Morphology of Czech).* Karolinum, Charles Univeristy Press, Prague, Czech Republic, 2004. **Available:** BibTeX

- Jan Hajič: "Morphological Tagging: Data vs. Dictionaries." In: *Proceedings of the 6th Applied Natural Language Processing and the 1st NAACL Conference, Seatle, Washington, U.S.A.*, 2000, pp. 94–101. **Available:** PDF, PS, BibTeX

- Jan Hajič, Barbora Vidová Hladká: "Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset." In: *Proceedings of the COLING–ACL Conference, Montreal, Canada*, 1998, pp. 483–490. **Available:** PDF, PS, BibTeX

- Barbora Vidová-Hladká: *Czech Language Tagging.* PhD thesis, ÚFAL MFF UK, Prague, Czech Republic, 2000. **Available:** PDF, PS, BibTeX

6.3.3 Parsing

- Jan Hajič, Barbora Hladká, Daniel Zeman, Michael Collins, Lance Ramshaw, Christoph Tillmann, Eric Brill, Douglas Jones, Cynthia Kuo, Ozren Schwartz: *Core Natural Language Processing Technology Applicable to Multiple Languages.* Center for Language and Speech Processing, Johns Hopkins University, Baltimore, USA, 1998. **Available:** PDF, PS, BibTeX

- Vladislav Kuboň: *Problems of Robust Parsing of Czech.* PhD thesis, ÚFAL MFF UK, 2001. **Available:** PDF, PS, BibTeX
- Ryan McDonald, Fernando Pereira, Kiril Ribarov, Jan Hajič: "Non-Projective Dependency Parsing using Spanning Tree Algorithms." In: (ed.): *Proceedings of Human Langauge Technology Conference and Conference on Empirical Methods in Natural Language Processing (HTL/EMNLP),*

  Vancouver, BC, Canada, Oct. 6-8, 2005, pp. 523–530. **Available:** PDF, PS, BibTeX
- Kiril Ribarov: *Automatic Building of a Dependency Tree–The Rule-Based Approach and Beyond.* PhD thesis, ÚFAL MFF UK, Prague, Czech Republic, 2004. **Available:** PDF, PS, BibTeX
- Daniel Zeman: *Parsing with a Statistical Dependency Model.* PhD thesis, ÚFAL MFF UK, Prague, Czech Republic, 2005. **Available:** PDF, PS, BibTeX

6.3.4 Automatic functor assignment

- Petr Sgall, Zdeněk Žabokrtský, Sašo Džeroski: "A Machine Learning Approach to Automatic Functor Assignment in the Prague Dependency Treebank." In: R. M. Rodríguez, C. Paz Suárez Araujo (eds.): *Proceedings of the 3rd International Conference on Language Resources and Evaluation, Las Palmas, Canary Islands, Spain*, (5), European Language Resources Association, 2002, pp. 1513–1520. **Available:** PDF, PS, BibTeX
- Zdeněk Žabokrtský: "Automatic Functor Assignment in the Prague Dependency Treebank." In: P. Sojka, I. Kopeček, K. Pala (eds.): *Proceedings of the 3rd International Conference on Text, Speech and Dialogue, Brno, Czech Republic*, Springer-Verlag Berlin Heidelberg New York, 2000, pp. 45–50. **Available:** PDF, PS, BibTeX
- Zdeněk Žabokrtský: "Automatic Functor Assignment in the Prague Dependency Treebank." In: *ÚFAL Technical Report*, 10, MFF UK, Prague, Czech Republic, 2001. **Available:** PDF, PS, BibTeX

Chapter 7

Distribution and license

For using PDT 2.0, you have to fill in the License form and sign it electronically (for an exception, see below). See the text of the License in Section 7.1.

There are two ways to get PDT 2.0. The standard way is to order the full PDT 2.0 distribution through the Linguistic Data Consortium at <http://www.ldc.upenn.edu>; during the ordering process, you will be redirected to the form-based License web page, which you have to fill in for your order to be completed.

The other option is to download a part of PDT 2.0 directly from our web pages at <http://ufal.mff.cuni.cz/pdt2.0>; it is an exact copy of the distribution provided by LDC, but only a small sample of the annotated data is included. You can do so before or after filling the registration form based on the License at http://ufal.mff.cuni.cz/corp-lic/pdt20-reg.html[1], but you are not allowed to *use* anything what you have downloaded (tools, sample data, etc.) without filling in the form. In other words, *this license is not valid until registration.*

Parts of the distribution might be covered by the GNU Public License (GPL). Such tools and data packages explicitly say so (they are typically available also from other sources, such as author's personal web pages and standard Open Source and GNU software repositories, e.g. from sourceforge.net). In such a case, the GPL has precedence over this license. If the material you have downloaded or are using consists *solely* of such GPL-covered tools and data packages, you are *not required* to register under this license; however, we would like you to do so anyway (even though in fact its rules and conditions do not materially affect you in such a case) to help us secure future funding by having as many registered users as possible.

7.1 License agreement

Research-Usage License Agreement for the Prague Dependency Treebank, version 2.0

between

```
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University
Malostranské náměstí 25
CZ-11800  Praha 1
Czech Republic
pdt@ufal.mff.cuni.cz
http://ufal.mff.cuni.cz/
```

(the Proprietor)

and

```
Name:
Institution:
```

[1] <http://ufal.ms.mff.cuni.cz/corp-lic/pdt20-reg.html>

```
Address (street, city, ZIP):
Country:
Telephone(s):
Fax(es):
E-mail:
```

(the User)
whereas

A The Prague Dependency Treebank version 2.0 (PDT 2.0) is a collection of textual data and documentation containing linguistic annotations and software tools for their processing as described in the documentation, developed at and by the Proprietor under the following support: Ministry of Education of the Czech Republic projects No. VS96151, LN00A063, 1P05ME752, MSM0021620838, and LC536, Grant Agency of the Czech Republic grants Nos. 405/96/0198, 405/96/K214 and 405/03/0913, research funds of the Faculty of Mathematics and Physics, Charles University, Czech Republic, Grant Agency of the Academy of Sciences of the Czech Republic No. 1ET101120503 and 1ET101120413, Grant Agency of the Charles University No. 489/04, 350/05, 352/05 and 375/05 and the U.S. NSF Grant #IIS9732388.

B The Proprietor is the copyright holder of PDT 2.0 and is entitled to grant a license to the User.

C The User is an academic, educational or research institution, or other organization, or an individual wishing to make use of PDT 2.0 for research and/or education purposes.

It is hereby agreed as follows:

1. This agreement is made on the *date of submission* and effective immediately.

2. The User is granted a non-exclusive license to use, modify, enlarge or enrich PDT 2.0 to extract information directly or indirectly in any form and volume, provided that PDT 2.0 itself or any derivative work is used only by the User her/himself or his/her immediate collaborators, employees, managers and/or her/his students from the same Institution for research purposes only, and provided she/he is continuously observing all the terms and conditions contained in this Agreement. If any part of PDT 2.0 contains its own license or an additional restriction, the more restrictive version and/or amendment of the license shall be in effect, unless specifically stated otherwise in such a part. In particular, all the documentation which is provided in either RTF, PDF or PostScript format should be considered a personal copy of the respective Author's reprint and handled as such.

3. The User shall not use PDT 2.0 itself or any derivative work (including but not limited to statistics obtained by using it or any derivative work thereof) based on it (however small the contribution of PDT 2.0 to such derivative work is) in whatever form for commercial purposes of any kind, nor for a deployment in any routinely used application, regardless whether it is of commercial nature or not.

4. The User shall include the following notice in all publications or publicly available materials, regardless of their form (printed, electronic, or other), describing work which uses PDT 2.0: "The Prague Dependency Treebank, version 2.0 has been developed by the Institute of Formal and Applied Linguistics, `http://ufal.mff.cuni.cz/`." In the case of printed materials, such as papers, journal articles, etc., one publication most suitable for reference with regard to User's work should be included from the list in the PDT 2.0 documentation; in the case of electronic publications on the Internet, the reference to the aforementioned web page should be included as a web link. Due to Proprietor's obligations with regard to the text copyright holders, text examples and citations from PDT 2.0 or any derivative work (regardless whether they include any annotations or not) are limited to 200 words per publication or series of publications on the same topic (whether printed, electronic, or in any other form).

5. The User agrees not to re-distribute or otherwise make publicly available PDT 2.0, or any derivative work based on it as described in paragraph 3, to a third party without a prior written permission of the Proprietor, with the exception of examples and citations as described in paragraph 4.

6. The User undertakes to adopt any security measures needed to protect the Proprietors' copyright in PDT 2.0 and undertakes to take all reasonable steps to ensure that no unauthorized use is made of PDT 2.0 and of any copies, derivative works or extracts thereof.

7. Any usage of PDT 2.0 which does not conform to the specification set forth in the 3rd paragraph of this Agreement (such as, e.g., commercial usage of PDT 2.0) is subject of separate negotiations and a written contract between the User and the Proprietor and/or other parties. The Proprietor is in general not obliged to enter and/or conclude such negotiations.

8. PDT 2.0 is provided as is. Therefore the Proprietor does not warrant the usefulness of PDT 2.0 for any purpose, regardless of formulations which can be found at some places in the accompanying documentation stating the intended purpose and use of PDT 2.0.

9. If the User reports to the Proprietor any discovered errors, inconsistencies or suggested corrections or improvements to PDT 2.0, the Proprietor undertake: (a) to maintain these comments in confidence and to use them only for the purposes of improving, modifying and/or maintaining PDT 2.0, (b) not to disclose the comments except in confidence to those of their employees or directors who need to know the same for the aforesaid purpose.

10. Should the Users by themselves or anyone acting on their behalf fail to comply with any of the conditions in this agreement (save with the written consent of the Proprietor) this agreement shall terminate immediately and PDT 2.0, its copies and derivative works based on it shall be destroyed at the User's site and at all sites under his control. Such termination shall be without prejudice to any claim which the Proprietor may have either for monies due and/or damages and/or otherwise.

11. Failure by the Proprietor to exercise or enforce any rights in this agreement shall not be deemed to be a waiver of any such right nor operate so as to bar the exercise or enforcement thereof at any time or times thereafter.

12. This agreement terminates if (a) the User destroys all copies of PDT 2.0 or any derivative work thereof, (b) the User or its Institution ceases to exist, unless all its obligations are transferred to a new entity, which is then considered to be bound by this Agreement. The User or its successor shall inform the Proprietor about any such transfer or succession; failure to do so will terminate this Agreement after one month after such transfer or succession. (c) the Proprietor ceases to exist without a legal successor.

13. The Proprietor shall keep the information about the User provided when submitting this Agreement in confidentiality and will not disclose it to other parties, except in a summary form such that individual users will not be identified, unless they specifically agree to such a disclosure in writing.

14. This agreement is governed by the laws of the Czech Republic and all disputes concerning this agreement will be resolved by its jurisdiction.

Chapter 8

Installation

In order to make the life of PDT users easier, we provide them with Linux and Windows installers. However, it should be noted that most PDT 2.0 components can be used directly from the CD-ROM or from its copy, or (some of them) can be installed separately by their own installers.

Installation on Linux. Launch the installer by executing the command **./Install-Linux.pl** in the root directory of the distribution. It will ask you to select the PDT components you wish to install, and to specify the target directory on your system. The components will be copied (and unpacked in some cases). Finally, you will be informed how to accomplish the installation of the tree editor `TrEd`.

Installation on Microsoft Windows. Launch the installer, e.g. by double-clicking the **Install-Windows** icon in the root directory of the distribution. At the very beginning, the installer checks whether Active State Perl is available on your system (Perl is necessary for functioning of the tree editor `TrEd`); if not, it informs you where you can download and install it from. Then the installer copies the selected components of the PDT 2.0 into the selected directory on your system (note that the Windows installer does not offer the installation of the chain of tools for automatic machine annotation, since they are based on Linux binaries). Finally, the separate Windows installer of the tree editor `TrEd` is executed.

The provided Linux and Microsoft Windows installers do not include the installation of `Netgraph`. If you want to install `Netgraph`, please consult the following documents:

- Netgraph Client Quick Installation: `doc/tools/netgraph/README_QUICK_INSTALL_CLIENT`
- Netgraph Server Quick Installation: `doc/tools/netgraph/README_QUICK_INSTALL_SERVER`
- Netgraph Server Installation Manual: `doc/tools/netgraph/netgraph_server_install.html`

Chapter 9

Credits

The following people have contributed directly one way or another to the creation and development of the Prague Dependency Treebank, version 2.0. Alphabetical order (based on their last names) is used throughout, except for publications (such as the Annotator's Guidelines) which respects the published order of the authors.

- **PDT 2.0**
  - **Morphological layer**
    - *Coordinator*: Barbora Hladká
    - *Linguistic Supervision*: Jan Hajič
    - *Annotation Manual*
      - *English version*: Daniel Zeman, Jan Hajič, Jiří Hana, Hana Hanová, Barbora Hladká, Emil Jeřábek
    - *Annotators*: Martin Buben, Jiří Hana, Hana Hanová, Emil Jeřábek, Lenka Kebortová, Kristýna Kupková, Pavel Květoň, Jiří Mírovský, Andrea Pfimpfrová
    - *Post-Annotation Checking*: Jiří Hana, Hana Hanová, Barbora Hladká, Emil Jeřábek
    - *Post-PDT 1.0 Checking*: Pavel Květoň, Petr Pajas, Pavel Pecina, Jan Štěpánek, Daniel Zeman, Zdeněk Žabokrtský
    - *Software and Technical Support*: Jan Hajič, Jiří Hana, Karel Skoupý
  - **Syntactical-analytical layer**
    - *Coordinator*: Jan Hajič
    - *Linguistic Supervision*: Jarmila Panevová
    - *Annotation Manual*
      - *Czech Version*: Alla Bémová, Eva Buráňová, Jan Hajič, Jiří Kárník, Petr Pajas, Jarmila Panevová, Zdeňka Urešová, Jan Štěpánek
      - *Translation to English*: Eva Hajičová, Zdeněk Kirschner, Petr Sgall
    - *Annotators*: Alla Bémová, Eva Buráňová, Jiří Kárník, Petr Pajas, Jan Štěpánek, Zdeňka Urešová
    - *Post-Annotation Checking*: Eva Buráňová, Jakub Dotlačil, Petr Pajas, Jan Štěpánek
    - *Post-PDT 1.0 Checking*: Petr Pajas, Jan Štěpánek, Zdeněk Žabokrtský
    - *Software and Technical Support*: Jan Hajič, Jiří Havelka, Michal Křen, Petr Pajas, Jan Štěpánek, Daniel Zeman
  - **Tectogrammatical layer**
    - *Coordinator*: Jan Hajič
    - *Linguistic Supervision*: Eva Hajičová, Jarmila Panevová, Petr Sgall
    - *Annotation Manual*
      - *Czech Version*: Marie Mikulová, Alla Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolářová-Řezníčková, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Magda Razímová, Petr Sgall, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, Zdeněk Žabokrtský

  - *Translation to English*: Alena Böhmová, Silvie Cinková, Eva Hajičová, Pavel Straňák
- *Annotators Training*: Veronika Kolářová-Řezníčková, Ivona Kučerová
- *Tectogrammatical Annotation Structure, Functor and Valency Frame Assignment*
  - *Coordinator*: Jan Hajič
  - *Annotators*: Alla Bémová, Eva Buráňová, Jakub Dotlačil, Marie Mikulová, Magda Razímová, Kateřina Součková, Zdeňka Urešová, Jana Vejvodová
  - *Post-Annotation Checking*: Václava Benešová, Ondřej Bojar, Jan Hajič, Markéta Lopatková, Petr Pajas, Jan Štěpánek, Zdeňka Urešová, Jana Vejvodová, Šárka Zikánová-Lešnerová, Zdeněk Žabokrtský
  - *Software and Technical Support*: Alena Böhmová, Petr Pajas, Jan Štěpánek, Zdeněk Žabokrtský
- *Topic-Focus Articulation*
  - *Coordinator*: Jiří Havelka
  - *Annotation Guidelines*: Kateřina Veselá
  - *Annotators*: Eva Buráňová, Anna Dostálová, Barbora Smrčková, Kateřina Veselá, Šárka Zikánová-Lešnerová
  - *Post-Annotation Checking*: Jakub Dotlačil, Jiří Havelka, Barbora Smrčková, Kateřina Součková, Kateřina Veselá, Šárka Zikánová-Lešnerová
  - *Software and Technical Support*: Jiří Havelka
- *Coreference*
  - *Coordinator*: Zdeněk Žabokrtský
  - *Annotation Guidelines*: Veronika Kolářová-Řezníčková, Lucie Kučová
  - *Annotators*: Kateřina Černá, Lucie Kučová, Jana Vejvodová
  - *Post-Annotation Checking*: Lucie Kučová, Petr Pajas, Magda Razímová, Jiří Semecký, Jan Štěpánek, Zdeněk Žabokrtský
  - *Software and Technical Support*: Oliver Čulo, Petr Pajas, Zdeněk Žabokrtský
- *Grammatemes*
  - *Coordinator*: Zdeněk Žabokrtský
  - *Annotation Guidelines*: Magda Razímová
  - *Annotators*: Kateřina Marková, Kamila Pacovská, Magda Razímová
  - *Software and Technical Support*: Daniel Zeman
- *PDT Vallex*
  - *Coordinator*: Petr Pajas
  - *Annotators*: Alla Bémová, Veronika Kolářová-Řezníčková, Markéta Lopatková, Zdeňka Urešová
  - *Post-Annotation Checking*: Alla Bémová, Jan Hajič, Veronika Kolářová-Řezníčková, Markéta Lopatková, Petr Pajas, Zdeňka Urešová
  - *Software and Technical Support*: Petr Pajas, Zdeněk Žabokrtský

- **TOOLS**
  - **TrEd** Petr Pajas
  - **NTrEd** Petr Pajas, Zdeněk Žabokrtský
  - **Netgraph** Jiří Mírovský, Roman Ondruška
  - **Segmentation and tokenization of Czech texts** Jan Hajič, Michal Křen
  - **Morphological Analyzer of Czech** Jan Hajič, Jaroslava Hlaváčová
  - **Tagger** Jan Hajič
  - **Parser** Michael Collins, Václav Honetschläger
  - **PDT Analytical Function Assignment** Petr Pajas, Zdeněk Žabokrtský
- **PUBLICATIONS**
  - *Collection, Formatting*: Barbora Hladká, Petr Homola, Jiří Semecký

- **CD-ROM, WEB DESIGN**
  - *Directory structure*: Václav Honetschläger, Zdeněk Žabokrtský
  - *Installation script*: Ondřej Bojar
  - *Validation*: Petr Podveský
  - *PDT guide editors*: Václav Honetschläger, Zdeněk Žabokrtský
  - *Booklet*: Alena Böhmová
  - *Web design*: Václav Honetschläger

Chapter 10

Acknowledgments

The Development of the Prague Dependency Treebank, version 2.0 has been supported by the following organizations, projects and sponsors:

- Ministry of Education of the Czech Republic[1] projects No. VS96151, LN00A063, 1P05ME752, MSM0021620838 and LC536,
- Grant Agency of the Czech Republic[2] grants Nos. 405/96/0198, 405/96/K214 and 405/03/0913,
- research funds of the Faculty of Mathematics and Physics[3],
- Charles University[4], Prague, Czech Republic,
- Grant Agency of the Czech Academy of Science, Prague, Czech Republic[5] projects No. 1ET101120503, 1ET101120413, and 1ET201120505,
- Grant Agency of the Charles University[6] No. 489/04, 350/05, 352/05 and 375/05,
- and the U.S. NSF[7] Grant #IIS9732388.

We thank our partners in the above projects, namely, the Institute of the Czech National Corpus[8] and the Institute of Theoretical and Computational Linguistics[9] of the Faculty of Arts, Charles University, Prague, the former for providing us with the original raw data and both for insights into the problems while discussing them during the nine years that lead to the publication of PDT 2.0. We are grateful to the text providers, namely, the Lidové Noviny Publishers, Mladá Fronta Dnes, the original publishers of the Českomoravský Profit journal and Vesmír, s. r. o., for allowing us to include their texts into the PDT distribution. We would also like to thank many others whose work has been used during the work on PDT 2.0, mainly the authors of zillions of wonderful tools available under the GPL or otherwise made available to us, from Linux to Perl to many tiny but time saving programming gems.

[1] <http://www.msmt.cz>

[2] <http://www.gacr.cz>

[3] <http://www.mff.cuni.cz>

[4] <http://www.cuni.cz>

[5] <http://www.cas.cz>

[6] <http://www.cuni.cz/UK-33.html>

[7] <http://www.nsf.gov>

[8] <http://ucnk.ff.cuni.cz/>

[9] <http://utkl.ff.cuni.cz/>