Complex Corpus Annotation:

The Prague Dependency Treebank

Jan Hajič

ÚFAL MFF UK
Charles University
Malostranské nám. 25
CZ-11800 Prague 1
Czech Republic
hajic@ufal.mff.cuni.cz

Abstract

The Prague Dependency Treebank (Hajič et al., 2001) is approaching the publication of its second version in which the tectogrammatical annotation is being added to the morphological and analytical (surface-syntactic) one. In this article, the Prague Dependency Treebank as a whole is being described, including its brief history. In this volume, there are three more papers with a detailed account of some of the most recently tackled phenomena occurring at the tectogrammatical level of annotation (Panevová and Lopatková, 2004, Cinková and Kolářová, 2004, and Urešová, 2004).

1 Introduction

The idea of the Prague Dependency Treebank does not really come from Prague: let us tell the story now. First of all, the original inspiration came from Philadelphia (where else?): in the early 90s, the availability of the Penn Treebank (Marcus et al., 1993) was a fascinating thing (to us at least). Then, at the European ACL Conference in Dublin in 1995, a small group of us "Praguians" met to discuss the possibility of such a treebank (based on the dependency framework, of course - what else!). We had no money and therefore no people to carry it on, but we decided to push the idea through the national Czech Grant Agency (even though it was clear we cannot really call it a "treebank"[1], since such a word was quite a "dirty" one, then), proposing at the same time another large grant for a Czech National Corpus together with several other colleagues from the country and a project called the Laboratory for Language Data (with the idea that in would be in this Laboratory where the annotation would in fact take place). Fortunately enough, we were awarded all these three projects[2] and in the fall of 1996, the project could begin at a full speed.

[1] We called it then "validation of a theory", without mentioning any figures regarding the number of words or sentences for which such "validation" would be performed.

[2] The project was started by support from the grant GAČR No. 405/96/0198 ("Formal specification of language structures"), and the annotation effort has been made possible by the grant GAČR No.

In present-day computational linguistics (CL), availability of annotated data (spoken utterances, written texts) is becoming a more and more important factor in any new developments. Apart from speech recognition, where statistical methods are almost exclusively *the* solution and where the data is a *conditio sine qua non*, textual data are being used for the training phase of various statistical methods solving many other problems in the field of CL. While there are many methods which use texts in their plain (or raw) form (for unsupervised training), (much) more accurate results may be obtained if annotated corpora are available. It is believed that syntax (and therefore, syntactic annotation) helps for subsequent processing in the direction of "language understanding" (or "comprehension").

With the increasing complexity of such tasks, the data annotation itself is a complex task. While tagged corpora (pioneered by Henry Kučera in the 60's) are now available for English and other languages, syntactically annotated corpora are rare. We decided to develop a similarly sized corpus of Czech with very "deep" and rich annotation scheme.

The textual data used for the task contains general newspaper articles (40%; including but not limited to politics, sports, culture, hobby, etc.), economic news and analyses (20%), popular science magazine (20%), and information technology texts (20%), all selected from the early collection of the Czech National Corpus.

2 The Prague Dependency Treebank Structure

The Prague Dependency Treebank (PDT) has a three-level structure (with tokenized text being taken as the input to the whole system). Full *morphological* annotation has been done on the lowest (first) *level*. The middle level deals with syntactic annotation using dependency syntax; it is called the *analytical level*. The highest level of annotation is the *tectogrammatical level*, or the level of linguistic meaning. We annotate the same text on all three levels, but the amount of annotated material decreases with the complexity of the levels[3].

3 The Morphological Level

On the morphological level, a tag and a lemma is assigned to each word form as identified in the input text. The annotation contains no (syntactic) structure; no attempt is even made to put together analytical verb forms, for example.

405/96/K214 and by the project of the Ministry of Education of the Czech Republic No. VS96151. Later, the work continued under the project called Center for Computational Linguistics (2000-2004), MSMT CR Project LN00A063. The development of some software tools used in this project has been supported by the grant GAČR No. 405/95/0190 and by the individual author's grant OSF RSS/HESP 1996/195.

[3] For various reasons, mainly technical: it has been experimentally proved (Zeman, 1998) that serially applied machine learning and statistical methods perform better if every step is trained on the true automatic output of the previous step rather than the manual one. In order to achieve this, there must be separate (additional) training data available for the preceding step, resulting in most data being necessary for the beginning (the first step) of the analysis, namely, morphology, and the least for the last one, the tectogrammatical analysis.

3.1.1 The Czech tag system

Czech is an inflectionally rich language. The full tag set contains currently 4712 tags (including morphological variants, which are being distinguished). We are using a positional tag system, the full description of which can be found in (Hajič, 2004).

We use 13 grammatical categories in the tag. For each category, one symbol is used at a fixed position in the tag string.

Cat.	Cat. Name	Description	Example values
1	POS	Part of Speech	A – adjective, R – preposition
2	SUBPOS	Detailed part of speech	s – passive participle, V – vocalized prep., Q – rel. pronoun
3	GENDER	Gender (grammatical, agreement)	I – masc. inanimate, N – neuter
4	NUMBER	Number (grammatical)	S – sing., D – dual
5	CASE	Case (or required case, for prep.)	1 – Nom., 3 – Dat., 7 – Instrumental
6	POSSGENDER	Possessive gender (owner's gender)	F – fem, M – masc. anim.
7	POSSNUMBER	Possessive number (owner's number)	S – singular, P – plural
8	PERSON	Person (verbs, pronouns)	1, 2, 3
9	TENSE	Tense (for participles, some exceptions)	R – past, F – future, P - present
10	GRADE	Degree of comparison (adjectives, adv.)	1 – positive, 3 - superlative
11	NEGATION	Negation prefix present	N - negated
12	VOICE	Voice (verbs)	A – active, P – passive
13	RESERVE1	Unused	
14	RESERVE2	Unused	
15	VAR	Variant, style, register, abbreviation, …	1 – variant, 6 – colloquial, 8 – abbr.

A short example[4] now presents a simple sentence as a sequence of annotated words:

Form (Czech)	*(Lit.)*	Tag
,	,	`Z:-------------`
že	*that*	`J,-------------`
litera	*the-letter*	`NNFS1-----A----`
výše	*above*	`Dg-------2A---1`
uvedené	*of-mentioned*	`AAFS2----1A----`
mezinárodní	*of-international*	`AAFS2----1A----`
smlouvy	*of-agreement*	`NNFS2-----A----`

[4] Example from the weekly journal Českomoravský profit, 10/1994.

mezi	*between*	**RR--7----------**
ČR	*Czech Rep.*	**NNFXX-----A---8**
a	*and*	**J^------------**
SR	*Slovakia*	**NNFXX-----A---8**
bude	*will*	**VB-S---3F-AA---**
mít	*have*	**Vf--------A----**
co	*pretty*	**TT------------**
nevidět	*soon*	**Vf--------N----**

Special symbols are used for combinations of values that are not easily distinguished, or the processing of which was simply left for the future. In most cases, we use the symbol 'X' for 'any value' in the particular grammatical category.
The lemma represents a unique identification of the word in the morphological dictionary. Usually, the customary dictionary base form (headword) is used as the identification string, extended (if necessary) by a dash and a number distinguishing it from its homographs. We use the following convention: all forms of a lemma must have the same part of speech, and for nouns, they also have to have the same gender. (This is, obviously, in accordance with the conventions of the morphological dictionary we use – see below in 3.1.2 Morphological Analysis).

3.1.2 Morphological analysis

Morphological analysis is a process the input of which is a word form as found in the text, and the output of which is a set of possible lemmas which represent such form in the dictionary, with each lemma accompanied by a set of possible tags (as defined in the previous section). For example, for the word form *ženu* the morphological analysis returns the following results:

Lemma	tag(s)
žena (*woman*)	**NNFS4-----A----**
hnát (*to rush*)	**VB-S---1P-AA---**

This example exhibits an ambiguity at the lemma level, but no ambiguity within the lemmas. On the other hand, the word form *učení* displays both types of ambiguity:

Lemma	tag(s)
učení (*theory*)	**NNNS1-----A----, NNNS2-----A----,** **NNNS3-----A----, NNNS4-----A----,** **NNNS5-----A----, NNNS6-----A----,** **NNNP1-----A----, NNNP2-----A----,** **NNNP4-----A----, NNNP5-----A----**
učený (*educated*)	**AAMP1----1A----, AAMP5----1A----**

There could be as many as five different lemmas for a given word form and as many as 27 different tags for one lemma.

Morphological analysis currently covers about a million Czech lemmas (including derivations), and is based on about 520,000 stems. It can recognize about 25 million word forms and their tags.

3.1.3 The process of manual morphological annotation

Morphological analysis is the first step towards the first level of annotation (morphological tagging) in the Prague Dependency Treebank. It can proceed fully automatically and very quickly (about 20000 word forms per second on today's average machine). We have developed a special software tool (called `sgd` on a Unix platform, and `DA` under Windows) which allows for an easy manual disambiguation of the morphological output. It also helps the annotators to edit the output of the morphology, thus allowing for identification of possible problems and unknown words in the morphology itself.

The morphological annotation has been performed on every sentence in the PDT twice, with a third person resolving the differences between the two annotators. The inter-annotator agreement has been around 97% (measured as the percentage of input tokens receiving the same tag by both annotators). After the adjudication process, there are still errors, though; at the present time, as we are preparing the version 2.0 of the PDT, we are able to better identify those errors (based on the upper levels of annotation) and we are correcting them.

A total of 1,800,000 words (tokens) is now available with manually annotated lemmas and tags.

3.2 The Analytical Level

The analytical (surface-syntactic) level of annotation is a newly designed level to more easily use (and compare) the results achieved in English parsing to Czech, and to have a preliminary analysis of a sentence structure before proceeding to the most detailed level, the tectogrammatical one. We have chosen the dependency structure to represent the syntactic relations within the sentence. The basic principles can be thus formulated as follows:

- The structure of the sentence is an oriented, acyclic graph with one entry (root) node; the nodes of the tree are annotated by complex symbols (attribute-value pairs);
- The number of nodes of the graph is equal to the number of words in the sentence plus one for the extra root node;
- The annotation result is only
    - 1. the *structure* of the tree,
    - 2. the *analytical function* of every node.

An analytical function determines the relation between the (dependent) node and its governing node (which is the node one level up the tree). All the other node attributes (see the table below) are used as guidance for the annotators, or they are used as an input or intermediate data for various automatic tools which participate in the annotation process, but are not considered to be the result of analytical annotation. In particular, the tags and lemmas are taken from the morphologically annotated data, and they are merged into the resulting data structure.

The first 10 node attributes are summarized in the following table (there are 8 more "technical" attributes used for macro programming as intermediate data holders etc.):

Attribute name	Brief description
`lemma`	lemma (see sect. 3, The Morphological Level)
`tag`	morphological categories, or tag (see sect. 3, The Morphological Level)
`form`	word form, after minor changes in some cases (error correction)
`afun`	the analytical function, or the type of dependency relation (towards the governing node)
`origf`	original word form as found in the text
`origap`	formatting (preceding the original word form)
`gap1,gap2,gap3`	formatting info preceding `form`, parts 1,2,3
`ord`	sequence no. of the word form in a sentence

The annotation rules are described in the manual (Bémová et al. 1997), the final version of which is available together with the annotated data (and much more) on the Prague Dependency Treebank v1.0 CD (Hajič et al., 2001).

These rules follow, where possible, the traditional grammar books, but are both extended (where no guidance has been found in such books) and modified (where the current grammars are inconsistent). They are intentionally as independent of any formal theory as possible (even though the decision to use the traditional - at least in Prague - dependency representations is certainly not quite theory independent - but in fact, this decision made our lives easier because of several phenomena inherently occurring in Czech (non-projective constructions, see e.g. Hajičová et al., 2004), which would otherwise result in the well-known "crossing brackets" problem).

In the following table, all possible values of the analytical function attribute (`afun`) are described briefly. The existence of a "suffixed" version (`_Co` for coordination, `_Ap` for apposition, `_Pa` for parenthetical expressions) is marked by an `x`.

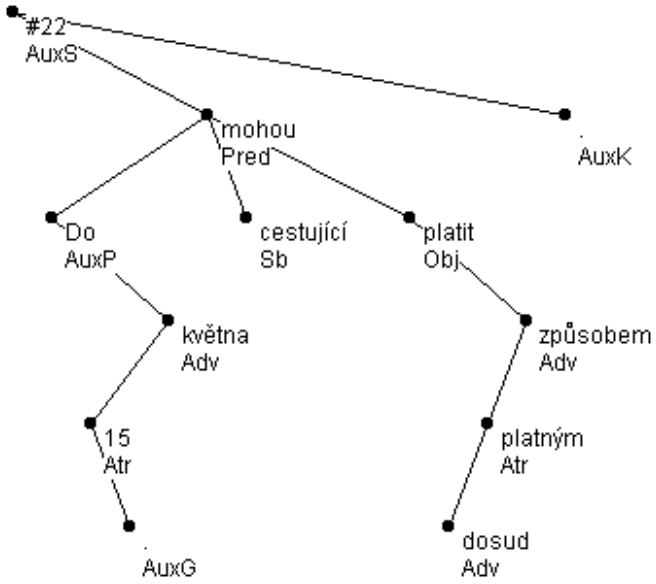
`afun`	`_Co`	`_Ap`	`_Pa`	Description
`Pred`	x	x	x	Predicate if it depends on the tree root (#)
`Sb`	x	x	x	Subject
`Obj`	x	x	x	Object

afun	_Co	_Ap	_Pa	Description
Adv	x	x	x	Adverbial (without a detailed type distinction)
Atv	x	x	x	Complement; technically depends on its non-verbal governor
AtvV	x	x	x	Complement, if only one governor is present (the verb)
Atr	x	x	x	Attribute
Pnom	x	x	x	Nominal predicate's nominal part, depends on the copula "to be"
AuxV	x	x	x	Auxiliary Verb "to be" (*být*)
Coord	x	x	x	Coordination, main node
Apos	x	x	x	Apposition, main node
AuxT	x	x	x	Reflexive particle *se*, lexically bound to its verb
AuxR	x	x	x	Reflexive particle *se*, which is neither Obj nor AuxT (passive)
AuxP	x	x	x	Preposition, or a part of compound preposition
AuxC	x	x	x	Conjunction (subordinate)
AuxO	x	x	x	(Superfluously) referring particle or emotional particle
AuxZ	x	x	x	Rhematizer or other node acting to stress another constituent
AuxX				Comma (but not the main coordinating comma)
AuxG				Other graphical symbols not classified as AuxK
AuxY	x	x	x	Other words, such as particles without a specific (syntactic) function, parts of lexical idioms, etc.
AuxS				The (artificially created) root of the tree (#)
AuxK				Punctuation at the end of a sentence or direct speech or citation clause
ExD	x	x	x	Ellipsis handling (Ex-Dependency): function for nodes which "pseudo-depend" on a node on which they would not depend if there were no ellipsis.
AtrAtr, AtrAdv, AdvAtr, AtrObj, ObjAtr	x	x	x	A node (analytical function: an attribute) which could depend also on its governor's governor (and have the appropriate other function). There must be no semantic or situational difference between the two cases (or more, in case of several attributes depending on each other). The order represents the annotator's preference, but is largely unimportant.

As an example of an analytical-level annotation of a sentence we present here the representation of the sentence

Do	*15.*	*května*	*mohou*	*cestující*	*platit*	*dosud*	*platným*	*způsobem*
Till	*15th*	*May*	*can*	*passengers*	*pay*	*hitherto*	*valid*	*way.*

(Until May 15, the passengers can pay in the way currently used.)



The original word forms as well as the attribute values of the analytical functions are displayed. This example shows

- the extra root node of the tree (showing the number of the sentence within a file)
- the handling of an analytical verb form (modal verb *mohou* + infinitive *platit*)
- the fact that the verb is the governing node of the whole sentence (or of every clause in compound sentences), as opposed to the complex subject - complex predicate distinction made even in the otherwise dependency-oriented traditional grammars of Czech, such as (Šmilauer 1969)
- attachment of a manner-type adverbial to an analytical verb form
- handling of a date expression
- prepositional phrase structure (preposition on top)

and, of course, all the analytical functions assigned to these nodes.

3.3 The Tectogrammatical Level

The tectogrammatical level of annotation is based on the framework of the Functional Generative Description (FGD) as it has been developed in Prague by Petr Sgall and his

collaborators since the beginning of the 1960's (for a most detailed and integrated formulation, see Sgall, Hajičová and Panevová 1986). The basic principles of annotation are different from those on the analytical level. Instead of requiring every word to become a node, we require that only every autosemantic word become a node. On the other hand, all nodes deleted on the surface - and thus on the analytical level – are added.

The tectogrammatical level is the most elaborated, complicated but also the most theoretically-based level of a semantico-syntactic (or ``deep syntactic'') representation. The tectogrammatical level annotation scheme is divided into four "sublevels" (or perhaps better, subareas, since they are all intertwined and do not form separate levels):

- dependencies and functional annotation,
- the topic/focus and deep word-order annotation,
- coreference, and
- the "deep" grammatical information.

As an additional data structure we use a syntactic lexicon, mainly capturing the notion of *valency*. The lexicon is not needed for the interpretation of the tectogrammatical representation itself,[5] but it is helpful when working on the annotation since it defines when a particular node that is missing on the surface should be created. In other words, the notion of (valency-based) ellipsis is defined by the dictionary. But before describing the dictionary, let us talk first about the core sublevel of annotation.

3.3.1 Dependencies and Functors

The tectogrammatical level goes beyond the surface structure of the sentence, replacing notions such as ``subject'' and ``object'' by notions like ``actor'', ``patient'', ``addressee'' etc. The representation itself still relies upon the language structure itself rather than on world knowledge. The nodes in the tectogrammatical tree are *autosemantic words* only[6]. Dependencies between nodes serve as the relations between the (autosemantic) words in a sentence, for the predicate as well as any other node in the sentence. The dependencies are labeled by *functors*[7], which describe the dependency relations. Every sentence is thus represented as a dependency tree, the nodes of which are autosemantic words, and the (labeled) edges name the dependencies between a dependent and a governor.

The dependency edge labels (functors) are much more detailed than the analytical functions (see the analytical function table in Sect. 3.2). They can be divided in several ways; here we use rather technical classification:

1. the separate root of the tree,
2. verbal and other complementations,
3. coordination, apposition and other functors for other "grouping" nodes,

[5] Nor for further analysis (say, a logical one) based on it, nor (in the other direction) for generation (synthesis) of surface sentences.
[6] By "autosemantic" we mean words that have lexical meaning, as opposed to just grammatical function.
[7] At two levels of detail; here we ignore so-called *subfunctors*, which provide the more detailed subclassification.

4. other functors that can be classified as neither describing autosemantic nor the "grouping" nodes.

We use over 80 different functors. In the following table, only the most important ones are described.

Functor class	Functor type	Description and examples
Root	Technical	SENT – Technical root of the tree
	Utterance root	PRED – Predicate of main clause in sentence DENOM – Nominal head of nominal expression
Dependency	Verbal Inner Participants	ACT – Actor PAT – Patient ADDR – Addressee ORIG – Origin EFF – Effect
	Time	TWHEN – When? TTILL – Till when? TSIN – Since when? TFHL – For how long? THL – How long? TFRWH – From when? TOWH – To when? TPAR – Parallel events THO – How often?
	Location	LOC – Location (non-directional) DIR1 – From where? DIR2 – Through where? DIR3 – To where?
	Manner	MANN – General manner MEANS – Means to achieve something RESL – Result REG – "with regard to", "according to" CRIT – Criterion or norm EXT – Extent ACMP – Accompaniment DIFF – Difference CPR – Comparison
	Implication	CAUS – Cause COND – Condition AIM – Aim INTT - Intention

Functor class	Functor type	Description and examples
	Other	BEN – Benefactor SUBS – Substitution HER – Heritage CONTRD - Contradiction RSTR – General attribute (of nouns) AUTH – Authorship APP – Appurtenance or property MAT – Material, container ID – Identity (name or description) COMPL – Complementizer (verb-noun "double dependency")
Grouping	Coordination	CONJ – Conjunction DISJ – Disjunction CONFR – Confrontation (clauses) CONTRA – Contrariety (expressions) GRAD – Gradation ADVS – Adversative CSQ – Consequence REAS – Reason OPER – Operand (mathematical-like expr.)
	Parenthesis	PAR – Root of parenthesis
	Rhematizer	RHEM – rhematizer (negation, only, also, …)
Other non-dependency		ATT – attitude PREC – Loose backward reference VOCAT – Addressing vocative expression PARTL – Unidentified particle, interjection INTF – Intensifier DPHR – Part of fixed phrase, idiom CPHR – Semantic part of light verb construct FPHR – Foreign language phrase CM – Part of conjunction

Many nodes found at the morphological and analytical levels disappear[8] (such as function words, prepositions, subordinate conjunctions, etc.). The information carried by the deleted nodes is not lost, of course: the relevant attributes of the autosemantic nodes they belong to now contain enough information to reconstruct them (even though such a reconstruction is not trivial, since it amounts to natural language generation from a semantic representation).

[8] Based on the principle of using only autosemantic words in the representation.

Ellipsis is being resolved at this level. Insertion of nodes is driven by the notion of *valency* (see below the section on Dictionary) and completeness (albeit not in its mathematical sense): if a word is deemed to be used in a context in which some of its valency frames applies, then all the frame's slots are to be "filled" (using regular dependency relations between nodes) by either existing nodes or by newly created nodes, and these nodes are annotated accordingly. Actual ellipsis (often found in coordination, direct speech etc.)[9] is resolved by creating a new node and copying all relevant information from its origin, keeping the reference as well.

Every node of the tree is furthermore annotated by such a set of grammatical features that enables to fully capture the meaning of the sentence (and therefore, to recover - at least in theory, see above the note of the NL generation problem – the original sentence or a sentence with synonymous linguistic meaning). The types of grammatemes belonging to individual nodes are defined by the notion of a *word class* (for autosemantic words, it corresponds to a "semantic class" of the word in question, i.e. semantic noun, verb, adjective or adverb). For example, (semantic) number is necessary to correctly form a sentence where no numeric expression is attached to a (semantic) noun. Other (obvious) example is (semantic) time: since auxiliaries are no longer present in the sentence structure, we have to have some means how to determine present, past or future tense (both relatively to the time when the sentence has been uttered or between clauses). Verbs do have other grammatemes, such as aspect, iterativeness, modalities of several types (related to modals such as "must" or "may", or to sentence modality: positive, interrogative, imperative sentence, etc.). Types of pronouns are also recorded where necessary.

3.3.2 The (syntactic) dictionary (valency lexicon)

The tectogrammatical level dictionary is viewed mainly as a valency dictionary of Czech (as theoretically defined in (Panevová, 1974, Panevová 1994); for recent account of the computational side and the actual dictionary creation, see Lopatková et al., 2002, Lopatková, 2003, Lopatková et al., 2003, Hajič et al., 2004, Žabokrtský and Lopatková, 2004) we mean the necessity and/or ability of (autosemantic) words to take other words as their dependents, as defined below.

Every dictionary entry is called a *lexia*, which may contain one or more *(valency) frames*. A frame consists of a set of *(valency) slots*. Each slot contains a *function* section (the actual *functor*, and an indication whether the functor is obligatory[10]), and an associated *form* section. The form section has no direct relation to the tectogrammatical representation, but it is an important link to the analytical level of annotation: it contains an (underspecified) analytical tree fragment that conforms to the analytical representation of a possible surface expression (or surface "realization", or simply "form") of the

[9] Nominal phrases, as used in headings, sports results, artifact names etc. are not considered incomplete sentences, even tough they do not contain a predicate; they are rather marked as denominalizations.
[10] By "obligatory" we mean that this functor (slot) must be present at the tectogrammatical level of annotation; this has immediate consequences for ellipsis annotation, cf. below.

particular slot. Often, the form section is as simple as a trivial (analytical) subtree with a single (analytical) dependency only, where the dependent node has a particular explicitly specified morphosyntactic case;[11] equally often, it takes the form of a two-edge subtree with two analytical dependencies: one for a preposition (together with its case subcategorization) as the dependent for the surface realization of the root of the lexia itself, and one for the preposition's dependent (which is completely underspecified). However, the form section can be a subtree of any complexity, as it might be the case for phrasal verbs with idiomatic expressions etc.

Moreover, the form section might be different for different expressions (surface realizations) of the lexia itself. For example, if the lexia is a verb and its surface realization is in the passive voice, the form of the (analytical) nodes corresponding to its (tectogrammatical) valency slots will be different than if realized in the active voice. However, relatively simple rules do exist to "convert" the active forms into the passive ones that work for most verbs; therefore, for such verbs, only the canonical (active) forms (by "form" we mean the analytical tree fragment as defined above) are associated with the corresponding valency slots. For irregular passivization problems there is always the possibility to enter the two (or more) different realizations explicitly into the dictionary. Many more rules have to be included, since passivization is not the only process that changes the form of a valency frame; most often, various expressions of modalities (or "near-modalities", that are not really treated as "true" modalities) have this effect.

A similar mechanism could be defined for nominalizations. Verbal nouns typically share the function section of the valency frame with their source verbs, but the form section might be a regular or an irregular transform of the corresponding form section. In the current version of the annotation valency lexicon, however, nouns (including verbal nouns) are given in full with their particular valency frame and its form.

Other issues are important in the design of the valency lexicon as well, such as reciprocity etc., but they are outside of the scope of this rather brief discussion.

The issue of word sense(s) is not really addressed in the valency dictionary. Two lexias might have exactly the same set of valency frames (as defined above, i.e., including the form section(s) of the slot(s)); in such a case, it is assumed that the two words have different lexical meaning (polysemy)[12]. It is rather practical to leave this possibility in the dictionary (however "dirty" this solution is from the puristically syntactic viewpoint), since it allows to link the lexias by a single reference to, e.g., the Czech WordNet senses (Pala and Smrž, 2004). The lexical (word sense) disambiguation problem is, however, being currently solved outside of the tectogrammatical level of annotation, even though eventually we plan to link the two, for obvious reasons. Then it will be possible to relate the lexias for one language to another in their respective (valency) dictionaries (at least

[11] Czech has seven morphosyntactic cases: nominative, genitive, dative, accusative, vocative, locative, and instrumental, usually numbered 1 to 7. In the example in the section 3.1.1, the case takes the 5th position in the positional representation of the morphological tag.

[12] On the other hand, it is clear that two lexias that do *not* share the same set of frames must have different lexical meaning as well, unless truly synonymous at a higher level of analysis.

for the majority of entries). From the point of view of machine translation, this can be viewed as an additional source of syntactically-based information of form correspondence between the two languages.
For more on the valency dictionary, see (Panevová and Lopatková, 2004, Cinková and Kolářová, 2004, and Urešová, 2004, in this volume).

3.3.3 Topic, Focus and Deep Word Order

Topic and focus (Hajičová, 2003, Hajičová et al., 2003) are marked, together with deep word order of the nodes of the tectogrammatical tree. The ordering of nodes is in general different from the surface word order, and all the resulting trees are projective by the definition of deep word order.

By *deep word order* (sometimes referred to as "contextual boundness") we mean such (partial) ordering of nodes at the tectogrammatical level that puts the "newest" information to the right, and the "oldest" information to the left, and all the rest in between, in the order of discourse-related notion of "newness". Such an ordering is fully defined at each single-level subtree of the tectogrammatical tree; i.e., all sister nodes *together with their head* are fully ordered left-to-right. The order is relative to the immediate head only; therefore, there exists such a total ordering of the whole tectogrammatical tree that the tree is projective. We believe that the deep word order is language-universal for every utterance in the same context, unless, roughly speaking, the structural differences are "too big" (or, in the case of translation, the corresponding translation is "too free").

In written Czech, the surface word order roughly corresponds to the deep word order (with the notable systematic exception of adjectival attributes to nouns, and some others), whereas the grammar of English syntax dictates in most cases a fixed order, and therefore the deep word order is often different (even though not always; even English has its means to shuffle words around to make the surface word order closer to the deep one, such as extraposition).

3.3.4 Co-reference

Grammatical and some textual co-reference relations are resolved and marked. Grammatical co-reference (such as the antecedent of "which", "whom", etc., control etc.) is simpler than the textual one (personal pronoun reference resolution etc.).

4 The Manual Annotation of the PDT

4.1 Organization

The manual tagging effort (level 1 annotation, see sect. 3) was coordinated by Barbora Vidová Hladká. She supervised a team of 5-7 students who double-tagged[13] the texts selected for the Prague Dependency Treebank. Each annotator has been given a description of the tag system (see sect. 3.1.1). Given that Czech moprhology is extensively taught at Czech high schools (both junior and senior), that's all they need from the linguistic point of view.[14] The discrepancy rate between any two annotators working on a single text is on average 5%, and there are virtually no opinion-type disagreements - the differences are human performance errors (typos, misunderstandings, etc.). The manual corrections of the annotated text revealed however that there are substantial differences among the annotators - ranging from 0.8 to 5% of errors. Other errors (about 1%, apart from missing words) were caused by errors made by the morphological analyzer during preprocessing. About 1,800,000 words have been annotated for PDT 1.0. The tools used for annotation are `sgd` (on Unix) and `DA` (for MS Windows), mutually compatible disambiguation programs with character-based window interface (see sect. 4.2.1).

Not surprisingly, the effort to organize the structural annotation (sect. 3.2) appeared to be a more complicated task than the organization of manual morphological annotation. There was little experience with such a task: we have learned from the LDC's experience with Penn Treebank, but there was no other description available of similar projects. The annotation itself begun in November 1996 by constituting a working group of 8 people, 5 of them hired just for the annotation of the data (the remaining three were faculty members). However, all the newly hired linguists were quite computer-literate, as were the computer science majors. Therefore their background allowed us virtually to skip any introduction to computational linguistics and we could start immediately with the annotation process itself.

The process of annotation has been (and still is) viewed as a cyclical process where the rules for annotation are being constructed on the basis of the evidence found in the data. Thus we have explained the basic principles of annotation to the annotators, and asked them to use existing grammar books, most notably (Šmilauer 1969), an old, but still the best Czech grammar description. This description builds also on a dependency framework, although there are some (easily identifiable and replaceable) deviations. We were aware of the fact that there are many gaps in such a traditional grammar from the point of view of an explicit annotation based on the basic principles stated above: mainly, the request to have each input word represented by a node in the tree (a request quite natural from the computational point of view) is largely not reflected in any human-

[13] Double-tagging means that the same text is processed twice by different annotators and the results are automatically compared and manually adjudicated to get a single (and presumably better) version.

[14] Eventually, a thin annotator's handbook as been developed as well, to solve certain technically difficult cases (such as foreign names, abbreviations, incomplete sentence with errors, etc.), mostly by convention.

oriented grammar description. Nevertheless, before starting to write authoritative guidelines based on such a grammar, we believed that a final version can be constructed on-the-fly with annotation corrections made later should the rules change.

The key software tool used was the GRAPH program, developed initially as an undergraduate thesis in 1995/96, and substantially enhanced afterwards (see also below, sect. 4.2). This tool allows for graphical viewing and editing of the dependency representation of annotated sentences.

All the annotators have helped to formulate the final wording in the Guidelines, and each of them is responsible for a certain section of the Guidelines (for example, for subject, or rhematizers and multiword units, etc.). Given their effort in this respect, and also their contribution to the formulation of the annotation rules during the first phase of the project, they all become not only the annotators, but also the authors of the Guidelines (Bémová et al. 1997).

In the end, 90,000 sentences (1.3 mil. word tokens) are available as part of the Prague Dependency Treebank at the end of the project. There were also other non-trivial tasks connected to the project: for example, the tagged data (level 1) had to be merged with the structurally annotated data, changes in morphology had to be incorporated, the resulting format was converted to SGML, etc. The PDT version 1.0 which contained the manually annotated data on the morphological and analytical levels was published in the fall of 2001 at the Linguistic Data Consortium in Philadelphia (Hajič et al., 2001).

The tectogrammatical-level annotation started in the year 2001. Preliminary guidelines have been used (published already as part of the PDT 1.0 CDROM). The annotators did not start from scratch this time: the analytical-level trees selected for tectogrammatical annotation have been preprocessed by a set of rules to decrease the annotation effort in cases where such rules can be formulated unambiguously, or for technical transformations of the tree that have been used by convention (Böhmová, 2001 and Böhmová and Hajičová, 2003). Later, after certain volume of the annotated data has been at our disposal, functor assignment has been rewritten to use a decision tree mechanism to further ease the task of manual functor assignment.

Based on the division of work into sublevels (see above in 3.3), the actual annotation has also proceeded along the four lines, with four groups (teams) working in parallel (some people participated in more than one effort). Also, a new platform-independent tool has been developed, called TrEd (Hajič, Hladká and Pajas, 2001), described in more detail below.

First, we have concentrated on the dependencies and functors, together with developing the valency dictionary and linking it to the corpus. Separately, exploratory work started for topic/focus and deep word order annotation, and for coreference annotation. The work on grammatemes have been postponed until 2003.

The corpus has been annotated only once (55,000 sentences total), with every fourth sentence double annotated (structure and functors) for inter-annotator agreement evaluation purposes. The valency dictionary has been developed by the annotators, sharing the dictionary among them during the course of the annotation. The structural

annotation was finished by mid-2003, and an 18-month checking and correction period ensued.

The newly developed annotation tool, data markup and sophisticated organization of the technical work allowed to work in parallel not only along the four major lines of annotation, but also within each line, to make changes and corrections relatively independently.[15] Those changes involve corrections after various automatic checks, merging the data from the four lines of annotation, corrections at the morphological and analytical levels (involving errors that were discovered during the tectogrammatical annotation and sometimes because of it), and many more things. The valency dictionary has been also "unified" by a single person, with changes mapped back to the data and manually corrected. Grammatemes have been filled in mostly automatically, based on quite sophisticated rules, even though some simplifications to their definitions had to be made to avoid the most time consuming annotation tasks.

4.2 Tools

Manual annotation does not mean that people are typing complicated formal representations by hand into a computer. Even the first annotation attempts in the times when graphical editing was resource-demanding and therefore not feasible were guided by software tools. These tools allowed the annotators to assign a formally correct entry only, avoiding expensive checking-and-correction process afterwards.

Based on the availability of computing power today, we decided that for the annotation of the PDT we should use as advanced tools as possible.

4.2.1 Morphological disambiguation: `sgd` and `DA`

We use a special purpose tool for morphological annotation, which allows for an easy disambiguation of lemmas and tags as output by the morphological analyzer. The tool has first been implemented under the Linux operating system under the name `sgd` (and is capable of running also on Solaris and other operating systems of the Unix type). It has been reimplemented also for the Windows platforms (under the `DA` name), to allow for annotators who did not have the possibility to install Linux on their home machines. The user interface is identical. The `sgd` tool is text-terminal based so it can be relatively easily (character coding problems aside) used from any `vt100`-capable terminal, as well as a from an `xterm` or similar programs.

The tools work full screen on texts in a SGML format (as defined by the Czech National Corpus' standard data type definition, namely, the `csts.dtd`) preprocessed by a morphological processor (see sect. 3.1.2 above). The annotators are presented with a list of ambiguous words as found in the input text (expandable to full text list, with ambiguous words marked by an asterisk). The full text context is also displayed in a separate window, with the active word marked by reverse video. The largest part of the

[15] Otherwise we would need a lot more time than those 18 months to finish the work.

screen is devoted to the disambiguation process itself. The annotator first chooses the correct lemma, and then, if needed (which is usually the case, as more than 45% words (tokens) are morphologically ambiguous in Czech), the correct tag. S/he has also the possibility to edit both the lemma and the tag, in case the morphological processor did not know the word altogether or made an error. The text is then saved with the lemmas and tags chosen by the annotators marked appropriately. There are other tools related to morphological annotation, but these are mostly standard Unix tools (diff, flex, awk, perl etc.). These help to resolve differences between two annotators on the same text and to do other conversions of the material.

4.2.2 The analytical level annotation tool: GRAPH

The analytical level, even though we are interested in the structure and one attribute (analytical function) "only", is a major challenge because of its inherently non-linear nature. We have used a program called rather uninspiratively GRAPH. This program works under Microsoft Windows (3.1 and 95) and has been developed as an undergraduate thesis based on initial specification developed long before the annotation project actually began. It has changed a lot since then - there were about 40 versions of it with bug fixes, minor and major updates. The program allows for drag-and-drop style editing of trees with annotated nodes. It is not just for dependency-based formal representations, even though it has special features (such as visual node ordering) which were inspired by such formalisms. Several files can be opened concurrently, (sub)trees may be copied among them using multiple-buffer clipboard, and files may be searched for node annotations. The display of trees (attributes to be displayed, colors, fonts, line thickness, etc.) is fully configurable to suit the task at hand as well as the annotator's preferences, which might depend on the hardware or other differences. The program can be completely mouseless driven, too.

One of the major features of the GRAPH program is the possibility to use macros - or in other words, the program is programmable. The programming language (which is interpreted at the moment) is similar to C but contains only those constructs necessary for the annotation tasks. The functions can be invoked interactively (by a keypress) or from the command line when starting the GRAPH program. These macros have been used so far for two different purposes:

- as shortcuts, asked for by the annotators, to avoid opening 2 or 3 menu windows when selecting the appropriate analytical function for a node in the tree;
- for a preliminary assignment of analytical functions to nodes when the tree structure is built, but before the manual node annotation.

The programming facility is not intended to be used by the annotators, but they are able to use the macros prepared by programmers. These macros can also be used for tree checking and transformations, if necessary e.g. after changes made in the annotation rules. The programming language allows for almost all the editing operations made normally by the annotators, including tree restructuring. Thus in principle, they could be used also for the initial tree structure assignment.

The shortcuts allow the annotators to assign an analytical function to an active node by a simple keypress, or a Ctrl and/or Shift plus a key in case of functions "suffixed" by `_Co`, `_Ap` or `_Pa`. These macros also store the value previously assigned to this node, and another macro function, when invoked, can thus revert to the previous value, should the annotator decide that s/he has made a mistake. There are also macros for node swap, for assignment of the `Atr` function to all nodes in a subtree (a frequent case near the leaves of the tree), and for special coordination and apposition handling.

The initial analytical function assignment was performed by an 800+ lines long function which tried to assign the most plausible analytical function to every node of a tree. The assignment was based on relatively simple hand-crafted rules. They were far from perfect, and sometimes intentionally disregarded some complicated contexts, but as the feedback from the annotators showed, they were correct in almost 80% cases. The initial assignment function could also be used (under a different name) on a file as a whole, which meant that the annotators did not have to run the macro on every tree. The batch feature of the GRAPH program also allowed to run the same macro on many files using a single command.

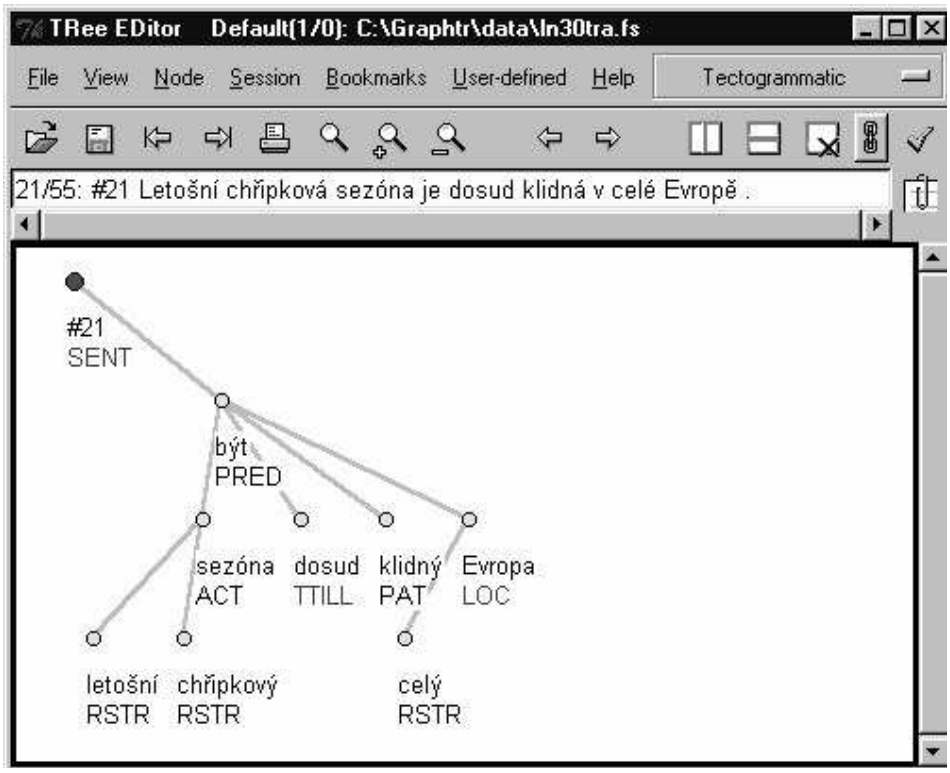
4.2.3 Tred: the tectogrammatical annotation tool

The graphical tree editor Tred has been developed originally when the final corrections and changes had been made to the analytical-level annotation before it was published. However, due to its advanced properties, easy extensibility and modularity and platform independence it has eventually been chosen as the main tool for the tectogrammatical annotation.

Tred is written in the **`perl`** programming language, it uses the **`perlTk`** extension for its graphical interface, and its basic functionality is the same as of the GRAPH tool (see above). It has been extensively used on both Linux and the Windows platforms. It can be customized for both the manual annotation work, as well as for batch processing of the annotated data. Thanks to its `perl` roots, it can be easily extended, and additional modules can equally easily be added for online[16] data processing, extending significantly the original idea of macros of the GRAPH editor. One of the extensions that has been heavily used is its lexical interface to the valency lexicon, which allows for both lexicon maintenance (adding, deleting, modifying entries and their associated valency frames) and for linking the lexical entries to the annotated data.[17] Tred also contains a general search interface that can be used both by the annotators as well as during the subsequent checking of the data; both simple and sophisticated searches (again, using `perl` expressions) can be launched.

The example below shows an open editing window with a tectogrammatical representation of the sentence "*This-year flu season is so-far quiet in [the] whole Europe.*"

[16] By "online" we mean during the manual annotation.
[17] The valency lexicon maintennace module can be also used outside of Tred as a stand-alone application.



Tred can display also additional links that are not part of the basic tree structure, in various graphic forms. It is used e.g. for coreference annotation, which links the consequent to the antecedent by a colored dashed arrow.

Two files can be displayed at the same time in two windows, side-by-side with differences automatically highlighted. This is used for visual checking of the double-annotated data, or different versions of the data. Also, the same sentence can be displayed on analytical and tectogrammatical levels, easing the comparison between the annotation of a particular sentence at these two levels.

5 Treebank Usage: Tagging and Parsing Unrestricted Text

The treebank can obviously be used for further linguistic research, as it contains a lot of material annotated in a way directly usable by original linguistic research, quickly searchable using different criteria. However, in the present contribution we will discuss a more “computational” usage of the treebank, namely, as a basis for creating a statistically-based tagger and a parser of unrestricted written text.

5.1 Full Morphological Tagging

We have developed a statistical model which has been successfully used for tagging (full morphological disambiguation), where it improved accuracy by 5 percentage points, from 80% (Hladká 1994, Hajič and Hladká 1997a) to 93% (Hajič and Hladká 1998, Hajič 2004) to 95% (Krbec et al. 2001). The statistical models are based on both the "classic" HMM:

$$p(T|W) = \prod_{i=1..n} p(t_i|t_{i-2},t_{i-1})\, p(w_i|t_i) \;/\; p(W)$$

where we use the Bayes formula to reverse the conditioning (simulating the well-known source-channel paradigm) and the trigram approximation for the tag language model, or the exponential probabilistic model of the form

$$p(y|x) = e^{\sum_{i=1..n} \lambda_i f_i(y,x)} / Z_\lambda(x)$$

where $f_i(y,x)$ is a feature selector function, which returns 1 or 0 depending on the value of y and the context x, λ_i is its weight, and $Z_\lambda(x)$ is a normalization factor making the distribution a probabilistic distribution which sums to 1.

The crucial property of this model, used successfully for many applications in tagging as well as in machine translation, is the set of *n* features (typically in the order of hundreds or thousands). These features are selected automatically, based on objective criteria, from a much larger "pool" of available features. The selection of features may be guided by two different principles: a "minimal cross-entropy" principle, which compares the probability distribution constructed to the training data (using the cross-entropy measure, or simply the probability of training data), or "minimal error rate" (again, on training data). We have chosen the second principle, as it more directly attacks the problem at hand.

The selection of features, however, depends also on the values of λ_i. The basic method for feature weight computation is the Maximum Entropy method. Unfortunately this method involves several numerical iterative algorithms which makes it rather slow. We believe, based on our experience with similar models (and with smoothing, which displays a similar "weighting" issue, in general) that the exact weight computation is not so important to the resulting model performance, and thus that the values of λ_i may be roughly - and quickly - approximated instead. This would allow us to select features from larger pools, thus enabling more sophisticated features to be selected.

5.2 Parsing

There are many attempts to parse sentences of natural language at various levels (Brill 1993a, Brill 1993b, Collins 1996, Collins 1997, Charniak 2000, Ribarov 1996). We aim here at syntactico-semantic parsing of unrestricted text. It is a well-known fact that hand-crafted rules work well for restricted domains and vocabularies, whereas they generally

fail for unrestricted text parsing. So far the (partial and imperfect, but still the best available) answer to this problem has been statistical parsing based on training on manually annotated data.

Having such a resource available for Czech (the Prague Dependency Treebank as described in the previous sections), we have successfully applied the Collins parsing model to Czech (Hajič et al., 1998, Collins et al., 1999). Collins parser currently achieves 82% dependency accuracy when trained on the PDT 1.0 analytical level training data. We also have at our disposal a modified version of the Charniak's parser for Czech (unpublished), which achieves slightly better performance (84% dependency accuracy when trained on the same data). Several other parsers have been developed since then, but none of them surpassed these two, except that (Zeman, 2004) constructed a combined "superparser" that shows the best result so far by combining several of the available parser outputs (having almost 85% accuracy for the best parse method). These parsers are complemented by a decision-tree implementation of function assignment that performs with about the same accuracy.

For tectogrammatical parsing, we currently use a set of manually written rules (Boehmová et al., 2003) that in fact requires the analytical parse be completed by either the Collins' or Charniak's parser, and then it transforms the analytical level tree to the tectogrammatical one. The result is worse than that on the analytical level, but we believe that it will improve once statistical methods are employed once the manual annotation at the tectogrammatical level is completed. The functor assignment is being performed by a mechanism similar to the analytical one, namely, a decision-tree functor classifier (implemented using the C5.0 software tool), with accuracy over 80%.

6 Conclusions

Building a treebank is an expensive and organizationally complicated task, especially when a rich annotation scheme is adopted such the one used in the Prague dependency treebank, where (roughly speaking) each word token from the selected text needs at least 15 attribute-value pairs to be filled in.

Everybody would certainly agree that to build a treebank is a difficult task. Our belief is, however, that all the hard work will pay off - in that not only us who are building it, but all the computational linguists interested in morphology and syntax of natural languages in general and of Czech or other inflectional and free word order languages in particular will benefit from its existence. The building of the treebank has been very fruitful even now, halfway through the whole treebank annotation: we have been effectively forced to describe the syntactic behavior of Czech more explicitly and more widely (in the sense of overall coverage, including also "peripheral" phenomena) than ever.

7 References

Bémová et al. (1997): Anotace na analytické rovině - příručka pro anotátory [Annotation on the Analytical Level - Annotator's Guidelines], Technical Report #4 (draft), LJD ÚFAL MFF UK, Prague, Czech Republic (in Czech).

Böhmová, Alena (2001): Automatic Procedures in Tectogrammatical Tagging. In PBML 76. MFF UK Prague.

Böhmová, Alena; Hajičová, Eva (2003): Large Language Data and the Degrees of Automation. In Proceedings of XVII International Congress of Linguists, CD-ROM. Matfyzpress, MFF UK Prague.

Brill, E. (1993a): Automatic Grammar Induction and Parsing Free Text: A Transformation-Based Approach. In: Proceedings of the 3rd International Workshop on Parsing Technologies, Tilburg, The Netherlands.

Brill, E. (1993b): Transformation-Based Error-Driven Parsing. In: Proceedings of the 12th National Conference on Artificial Intelligence.

Cinková, S. - Kolářová, V. (2004): Nouns as Components of Support Verb Constructions in the Prague Dependency Treebank. This volume.

Collins, M. (1996): A New Statistical Parser Based on Bigram Lexical Dependencies, In: Proceedings of the 34th Annual Meeting of the ACL'96, Santa Cruz, CA, USA, June 24-27, pp. 184-191.

Collins, M. (1997): Three Generative, Lexicalised Models for Statistical Parsing, In: Proceedings of the 35th Annual Meeting of the ACL/EACL'97, Madrid, Spain, pp. 16-23.

Hajič, Jan (2004): Disambiguation of Rich Inflection. Karolinum, Charles University Press, Prague. 332pp.

Hajič, Jan; Collins, Michael; Ramshaw, Lance; Tillmann, Christoph (1999): A Statistical Parser for Czech. In the Proceedings of ACL'99, Maryland, USA.

Hajič, J., and Hladká, B. (1997a): Probabilistic and Rule-based Tagger of an Inflective Language - A Comparison, In: Proceedings of the 5th Conference on Applied Natural Language Processing, ACL, Washington, DC, USA, pp. 111-118

Hajič, J., and Hladká, B. (1998): Morfologické značkování korpusu českých textů stochastickou metodou [Morphological tagging of Czech corpora using stochastic metods], In: Slovo a Slovesnost, Vol. 58, No. 4, ÚJČ AV ČR, Prague.

Hajič, Jan; Vidová-Hladká, Barbora; Pajas, Petr (2001): The Prague Dependency Treebank: Annotation Structure and Support. In Proceeding of the IRCS Workshop on Linguistic Databases Univesity of Pennsylvania, Philadelphia, USA, pp. 105-114.

Hajič, J., and Ribarov, K. (1997): Rule-Based Dependencies, In: Proceedings of the Workshop on Empirical Learning of Natural Language Processing Tasks, MLNet, Prague, Czech Republic, April 23-25, pp. 125-136

Hajič, Jan; Panevová, Jarmila; Urešová, Zdeňka; Bémová, Alevtina; Kolářová, Veronika; Pajas, Petr (2003): PDT-VALLEX: Creating a Large-coverage Valency Lexicon for Treebank Annotation. In Proceedings of The Second Workshop on Treebanks and Linguistic Theories, pp. 57--68. Vaxjo University Press.

Hajič, Jan; Vidová Hladká, Barbora; Panevová, Jarmila; Hajičová, Eva; Sgall, Petr; Pajas, Petr (2001): Prague Dependency Treebank 1.0. CDROM. CAT: LDC2001T10, ISBN 1-58563-212-0. Linguistic Data Consortium, Univ. of Pennsylvania, Philadelphia, USA. Also at http://ufal.mff.cuni.cz/pdt.

Hajičová, Eva (2003): Information structure and syntactic complexity. In Investigations into formal Slavic linguistics, pp. 169-180. Peter Lang.

Hajičová, Eva; Sgall, Petr; Veselá, Kateřina (2003): Information structure and contrastive topic. In Formal approaches to Slavic linguistics. The Amherst Meeting 2002, pp. 219-234. Michigan Slavic Publications.

Hajičová, Eva; Havelka, Jiří; Sgall, Petr; Veselá, Kateřina; Zeman, Daniel (2004): Issues of Projectivity in the Prague Dependency Treebank. In Prague Bulletin of Mathematical Linguistics MFF UK (in press).

Hladká, B. (1994): Programové vybavení pro zpracování velkých českých textových korpusů [Software for Large Czech Corpora Annotation], MSc thesis, MFF UK, Prague, Czech Republic.

Lopatková, Markéta (2003): Valency in the Prague Dependency Treebank: Building the Valency Lexicon. In Prague Bulletin of Mathematical Linguistics, pp. 37-60. MFF UK.

Lopatková, M., Panevová, J. (2004): Recent developments of the theory of valency in the light of the Prague Dependency Treebank. This volume.

Lopatková, Markéta; Řezníčková, Veronika; Žabokrtský, Zdeněk (2002): Valency Lexicon for Czech: from Verbs to Nouns. In Text, Speech and Dialogue. 5th International Conference, TSD 2002, pp. 147--150. Springer.

Lopatková, Markéta; Žabokrtský, Zdeněk; Skwarska, Karolina; Benešová, Václava(2003): VALLEX 1.0 Valency Lexicon of Czech Verbs. MFF UK.

Marcus, M.P., B. Santorini, and M. Marcinkiewicz (1993): "*Building a large annotated corpus of English: the Penn Treebank*," Computational Linguistics, vol. 19, pp. 313-330.

Panevová, J. (1974), On Verbal Frames in Functional Generative Description. Part I, Prague Bulletin of Mathematical Linguistics 22, 3-40, Part II, Prague Bulletin of Mathematical Linguistics 23, 1975, 17-52.

Panevová, J. (1994), Valency Frames and the Meaning of the Sentence. In: The Prague School of Structural and Functional Linguistics (ed. by Ph. L. Luelsdorff), Linguistic and

Literary Studies in Eastern Europe 41, Amsterdam-Philadelphia: John Benjamins, 223-243.

Ribarov, K. (1996): Automatická tvorba gramatiky přirozeného jazyka [The Automatic Creation of a Grammar of a Natural Language], MSc thesis, MFF UK Prague.

Řezníčková, Veronika (2003): Czech Deverbal Nouns: Issues of Their Valency in Linear and Dependency Corpora. In Proceedings of the Workshop on Shallow Processing of Large Corpora (SProLaC 2003), pp. 88--97. UCREL, Lancaster University.

Pala, K., Smrž, P. (2004): Building Czech Wordnet. Romanian Journal of Information Science and Technology Special Issue. Ed. By D. Tufis. Vol. 7, No. 1-2. pp. 79-88.

Sgall, P. et al. (1986): The Meaning of the Sentence and Its Semantic and Pragmatic Aspects, Reidel Publishing Company, Dordrecht, Netherlands; Academia, Prague, Czech Republic.

Šmilauer, V. (1947), Novočeská skladba [Syntax of Contemporary Czech], 1st ed., Prague.

Šmilauer, V. (1969), Novočeská skladba [Syntax of Contemporary Czech], 3rd ed., SPN, Prague, 574 pp.

Urešová, Z. (2004): The verbal valency in the Prague Dependency Treebank from the annotator's point of view. This volume.

Zeman, D. (2004): Parsing with a statistical dependency model. PhD Thesis. MFF UK Prague. In print.

Žabokrtský, Zdeněk; Lopatková, Markéta (2004): Valency Frames of Czech Verbs in VALLEX 1.0. In Frontiers in Corpus Annotation. Proceedings of the Workshop of the HLT/NAACL Conference, pp. 70-77.

Abstrakt (česky)

Pražský závislostní korpus (PDT, Hajič et al., 2001) obsahuje bohatou morfologickou, syntaktickou a syntakticko-sémantickou informaci ve formě manuálně provedené anotace. V tomto článku představujeme stručný popis celého PDT včetně seznamů hlavních značek užitých pro anotaci na jednotlivých rovinách. Rovněž jsou uvedeny některé zkušenosti z průběhu anotace, a jsou popsány i nástroje, které byly při anotaci použity. Na závěr článku uvádíme možnosti využití manuálně anotovaných korpusů pro vytváření automatických programových nástrojů pro analýzu jazyka na morfologické a syntaktické rovině.