Czech treebanking unlimited

Tomáš Jelínek, Vladimír Petkevič, Alexandr Rosen and Hana Skoumalová

Charles University, Faculty of Arts, Institute of Theoretical and Computational Linguistics
Celetná 13, 110 00 Praha 1, Czech Republic
{tomas.jelinek|vladimir petkevic|alexandr.rosen|hana.skoumalova}@ff.cuni.cz

Abstract

We build a large treebank of Czech, avoiding manual effort by using a parser, supplemented by a rule-based correction tool. A potentially underspecified morphological and syntactic annotation scheme offers multiple visualisation and export options, customizable in shape and detail according to the preferences of humans or computer applications. The annotation scheme consists of three layers: graphemics, morphology and constituency-based syntax, and is supported by a lexicon (with a morphological, multi-word and syntactic part) and a grammar. Annotation on any of the interlinked layers can be missing; ambiguous or undecidable phenomena are represented by underspecification and distributive disjunction.

1. Introduction

Treebanks are often built with considerable manual effort and cannot match the size of other text corpora. Yet whenever a language is lucky to have one, human users and applications alike benefit from its existence.[1] At first, the size of the corpus, the style and theoretical background of the annotation, or even its detail are not an issue. But sooner or later the users realize that at least for some tasks a treebank should reach higher volumes.[2] Additionally, various wishes concerning a "proper" annotation scheme are voiced, often reflecting the division of syntactic theory into a number of camps. Finally, some data are difficult to annotate unambiguously and call for underspecified description (Oliva, 2001). We wish to propose answers to these issues.

The bigger the better, but not at an unbearable decrease in reliability. The largest treebanks reach the modest sizes of several million words. To match the size of a balanced POS-tagged corpus, the use of parsing tools is inevitable. But parsers still perform less reliably than POS taggers and the cost of manual checking is prohibitive.

Building on previous efforts in treebank annotation of Czech, especially the Prague Dependency Treebank – PDT (Hajič, 2006, i.a.), we combine a stochastic parser (Holan and Žabokrtský, 2006) with a rule-based correction module (Jelínek, 2012), diminishing the parser's error rate. The goal is to provide syntactic annotation for all contemporary written texts in the Czech National Corpus.[3] The texts represent a balanced mix of genres, with the total of 1.3 billion tokens at the moment.

We are not alone in realizing the usefulness of "parse-banks". In addition to those serving as a testbed for a specific rule-based grammar and linguistic theory, such as LinGO Redwoods,[4] there are other large automatically parsed corpora available, such as LASSY,[5] with Dutch texts numbering 1.5 billion tokens. The higher error rate does not seem to be a serious obstacle for tasks such as automatic valency acquisition.

An additional way to make the automatically parsed corpus useful is to offer fully customizable visualisation and export, including various folding and filtering options. Then the results are potentially less sensitive to errors in details or in more embedded structures. The scheme can also represent inherent ambiguities (in principle impossible to resolve even in a wider context), and allow for uncertainty or underspecification, potentially remediable later.

Even more importantly, there are various modes of displaying syntactic structure, according to the preference of the user, who may subscribe to a specific syntactic theory and may be put off by theoretical bias of a treebank. Yet despite differences in appearance and focus, all linguistic theories strive to describe the same object – a natural language. There is a large pool of implicit wisdom shared by all syntactic theories and a significant overlap of knowledge can be extracted from all theory-specific formats.

This multitude of syntactic paradigms can be approached by introducing parallel annotation layers (Hautli et al., 2012). However, we opt for a more flexible scheme, which is feasible precisely because all theories have to cope with the same issues. We propose a treebank offering different views of syntactic annotation while based on a single core pattern. In addition to constituency and dependency trees of various shapes, suited to the taste of linguistic experts, some views may be appealing to a wider audience of students and professionals dealing with language.

With these aims in mind, our explicitly defined annotation scheme consists of a potentially underspecified morphological and syntactic core, complemented by multiple interaction shells, customizable in shape and detail according to the preferences of humans or computer applications.

In Section 2. we explore the annotation scheme in more de-

[1]There is ample evidence of the usefulness of treebanks, even of those concerning less widely spoken languages, such as Czech, cf. `http://ufal.mff.cuni.cz/pdt2.0/`.

[2]This may be a concern for experts in machine learning and linguistic theoreticians alike. The latter are sometimes interested in very specific and rare cases to support a theoretical claim. Due to the lack of such evidence in treebanks of a smaller size, they resort to elicitation and/or use a larger corpus without syntactic annotation (Przepiórkowski and Rosen, 2005).

[3]The "SYN" family of subcorpora, see `http://www.korpus.cz/english/struktura.php`.

[4]`http://lingo.stanford.edu/redwoods/`

[5]`http://www.let.rug.nl/~vannoord/Lassy/`

tail, before discussing the conversion from the dependency trees produced by the parser to our format in Section 3. and the rule-based correction module, applied to the parser's output, in Section 4.

2. The core structure

The texts are annotated by a single core scheme, interpretable in different ways. The syntactic structure part is based on constituency, which supports structural ambiguity and underspecification more easily and can serve as a source from which multiple representations can be derived (see Section 2.3.). The annotation is licensed by a formal description, a de facto grammar, which additionally facilitates format conversions (see Section 2.2.).

2.1. Separation of graphemics, morphology and syntax

Word order and syntactic structure are represented in the core structures as formally distinct dimensions to support the choice of similarly separate or integral visualisation and comparison. In fact, each sentence is represented at three inter-linked levels: graphemics (orthographic words), morphology (syntactic words), and syntax (trees). The level of graphemics allows for handling contractions and similar purely orthographical phenomena. Reflexives subject to haplology are restored 2.1., and contractions such as *ses*, represented as a single graphemic unit, are analysed as two morphological forms: here as a reflexive pronoun/particle and the 2nd person auxiliary. More mismatches in the number of tokens occur between the levels of morphemics and syntax, where punctuation is omitted.

(1)	Rozhodl	se	umýt.
	decided$_{masc,sg}$	REFL	wash$_{inf}$
	'He has decided to wash himself.'		

The haplologized item *se* is both a reflexive particle, a part of an inherent reflexive *rozhodl se*, and a reflexive pronoun as the object of the transitive verb *umýt se*. As such, it is represented as two tokens on the level of morphemics:

GRAPHEMICS	rozhodl	se		umýt
MORPHEMICS	*lpple,masc,sg*	*pcle*	*prnrefl,acc*	*inf*

The two interpretations of *se* appear as two nodes in the syntactic structure below. The boxed constituent stands for the inherently reflexive verb as a multiword.

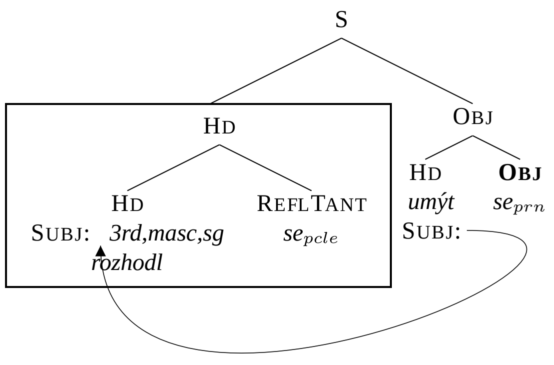

Figure 1: Haplology in a tree

Mismatches in the number of nodes at the individual levels (as in the case of *se* above) are kept at a minimum, elided items of all sorts are not restored as separate nodes but recorded in the node-internal structure of their heads or referring expressions as arguments, adjuncts or antecedents. E.g., in Fig. 1, PRO is represented equivalently as a link between the subject of the infinitive and the subject of the finite verb. All such phenomena are represented by linking the infinitive, predicative complement, base coordinated verb etc. across the structure with its argument. The link is labeled by the relevant syntactic function.

Links of another type make sure that agreeing categories in subject-predicate or adjective-noun agreement structures share identical values and the agreeing forms are identified. In the linear display (2), agreeing forms are underlined. Such links are predictable from syntactic structure and functional labels, and are inferred using the formal description of the annotation (grammar and lexicon).

Depending on the choice of the user, discontinuous (non-projective) structures can be represented as such, with crossing branches of the syntax tree, or made continuous (projective) on the syntactic level, with the order of the terminal nodes different from the lower levels. The parser is able to identify non-projectivity in the assumed dependency structures, and its results are subject to checks and modifications by correction rules.

2.2. Formal description of the annotation

To enforce consistency in the annotated data and to support interaction with the annotation, all syntactic structures in the corpus have to be licensed by a formal grammar. This includes a requirement that words and constituents have their appropriate (potentially underspecified) sets of features. A lexicon, compiled from existing resources and the corpus, and coupled with the grammar, is used to index word tokens using lemmas with appropriate categories, as well as compound forms and multi-word lexical units.

The grammar, based on the formalism of HPSG (Pollard and Sag, 1994), consists of (i) a definition of formal objects used to represent words and constituents, together with their properties, and (ii) constraints on the setup of larger objects, such as phrases, consisting of other objects. The formalism of HPSG allows for an arbitrary level of specificity of the description, and for including information of all kinds, which is useful for two reasons: a) The description should cover all structures and phenomena in the treebank, which is a hard task for a standard rule-based grammar. We need to relax some constraints or leave them unspecified, allowing for a prospect of a long-term grammar development. b) The core structures should be interpretable in various ways and according to various linguistic theories, relying on information in the structures.

Although the data in the corpus have a three-layered structure technically, they are compatible with the HPSG data format, where a single object provides all information about any expression – word or phrase. The information from the three levels, from the grammar and from the lexicon is merged and presented in a single virtual object.

In addition to checking the treebank's consistency, the grammar with the lexicon is useful (i) for inferring agreement relations, which do not have to be present in the input or supplied manually, (ii) as a base for describing format conversions to customized visualisation and export, and (iii) to supply additional annotation, such as parallel interpretations in prototypical cases of syntactic ambiguity.

The lexicon consists of three parts: morphological (specifying inflection paradigms of individual lemmas), syntactic (specifying their syntactic properties) and a lexicon of multi-word units, identifying analytical verb forms, potentially discontinuous collocations and phrasemes.

The setup of lexical resources and their relation to the three layers of annotation is shown in Fig. 2. Details about the tricky case of haplology are illustrated in Fig. 3, where tokens on the layer of graphemics are linked with tags on the morphological layer (such as `TT`... for reflexive particle, or `P7-X4`... for reflexive pronoun), which, in turn, are linked to lemmas in the morphological lexicon, and to nodes on the syntactic layer. The nodes point to the syntactic lexicon, including, e.g., info on valency, and – where appropriate – also to the multi-word lexicon.

2.3. Multiple options to display syntactic structure

The syntactic annotation scheme is designed to offer different views of its content, including constituency or dependency trees with a customizable level of abstraction (concerning, i.a., deep or surface dependencies, interpretation of function words, and identification of complex verb forms including inherent reflexives), and visualized in a horizontal or vertical mode with an arbitrary amount of detail, not necessarily by tree graphs. A linear display identifying the major (possibly discontinuous) constituents of a clause could be the option of choice for many users, see (2).

(2) A linear display of elementary syntactic structure:[6]
Ty by *se*s byl ušpinil.

Displayed information need not be explicitly encoded in the core structure, e.g., standard syntactic categories such as S, NP or PP are derived from the POS of the word or of the constituent's head and its valency.

An important side effect of less detailed visualisation is that some annotation errors can remain hidden. The parser produces about twice as many errors in syntactic structure than in syntactic functions, while the distributions of errors is uniform across all branches within a tree. By displaying only partial syntactic analysis, e.g., by using a combination of filters on main clause constituents, constituent labels, syntactic functions or syntactic structure, the number of errors visible by the user may be significantly reduced.

2.4. Surface and deep structure

Every constituent has a type – headed or unheaded – and a syntactic function. The list of types and functions is presented in Tables 1 and 2 below.

As Table 2 shows, a head can be distinguished as surface or deep; a function word such as preposition or verbal auxiliary is labeled as surface head while its sister is the deep

Label	Description
HEADED	Headed type
UNHEADED	Unheaded type with three subtypes:
– COORD	coordination structure
– ADORD	adordination structure
– UNSPEC	unspecified: other type of structure

Table 1: Types of constituents

Label	Description
SHD	Surface head
DHD	Deep head
HD	Head (simultaneously surface and deep)
SUBJ	Subject
OBJADVB	Object or Adverbial with two subtypes:
– OBJ	Object
– ADVB	Adverbial
ATTR	Attribute
VBATTR	Verbal complement
REFLTANT	Reflexive particle
DEAGENT	Reflexive particle with deagentive meaning
APOS	Apposition
INDEP	Independent constituent (parenthesis, noun in the vocative, etc.)
MEMB	Member of one of the unheaded subtypes

Table 2: Syntactic functions

head.[7] This allows for extracting both surface and deep dependencies from a single structure, see Fig. 4. Coordination and similar constructions are treated as headless (they are of the type UNHEADED).

(3)

Ty	by	ses	byl	ušpinil.
you	would	REFL+AUX$_{2nd,sg}$	be$_{pple}$	get dirty$_{pple}$

'You would have got dirty.'

The three structures in Fig. 4 are all possible renderings of a single analysis of (3). The constituent structure has function labels for subject, object, head, surface head and deep head, and it is followed by the derived surface and deep dependency structures.[8] Complex verb forms are highlighted by boldface, contractions by a box.

It is relatively straightforward to distinguish the three types of head, and thus the shape of the surface and deep dependency structure. Lexemes identifiable in a proper syntactic context as surface heads are labeled with specific syntactic functions by the parser and form a closed class. This distinction, implying the assignment of functional labels to other nodes in the vicinity, is performed by rules operating during the conversion of the parser output.

2.5. Ambiguity and partial information

Most corpora are annotated in an unambiguous way. Yet ambiguity is sometimes inevitable for fundamental reasons,

[6]The intended meaning of the text attributes is as follows: **subject**, predicate, *object*, agreeing forms.

[7]Przepiórkowski (2007) distinguishes *syntactic* and *semantic* heads in syntactic annotation of a corpus of Polish.

[8]For technical reasons, the labels mark nodes rather than edges, representing both constituency and functional relations. The nodes refer to categorial information appropriate to words or phrases, as in the analysis of (6) above.

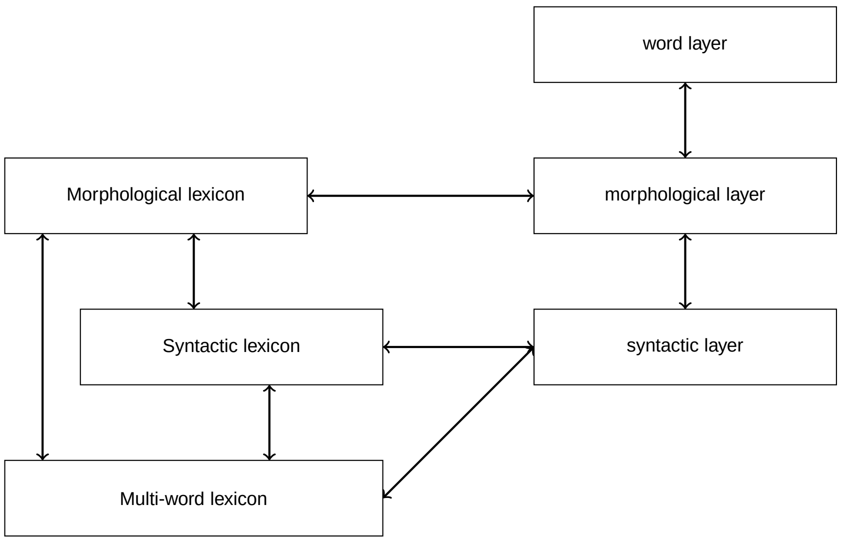

Figure 2: The setup of layers and lexical resources

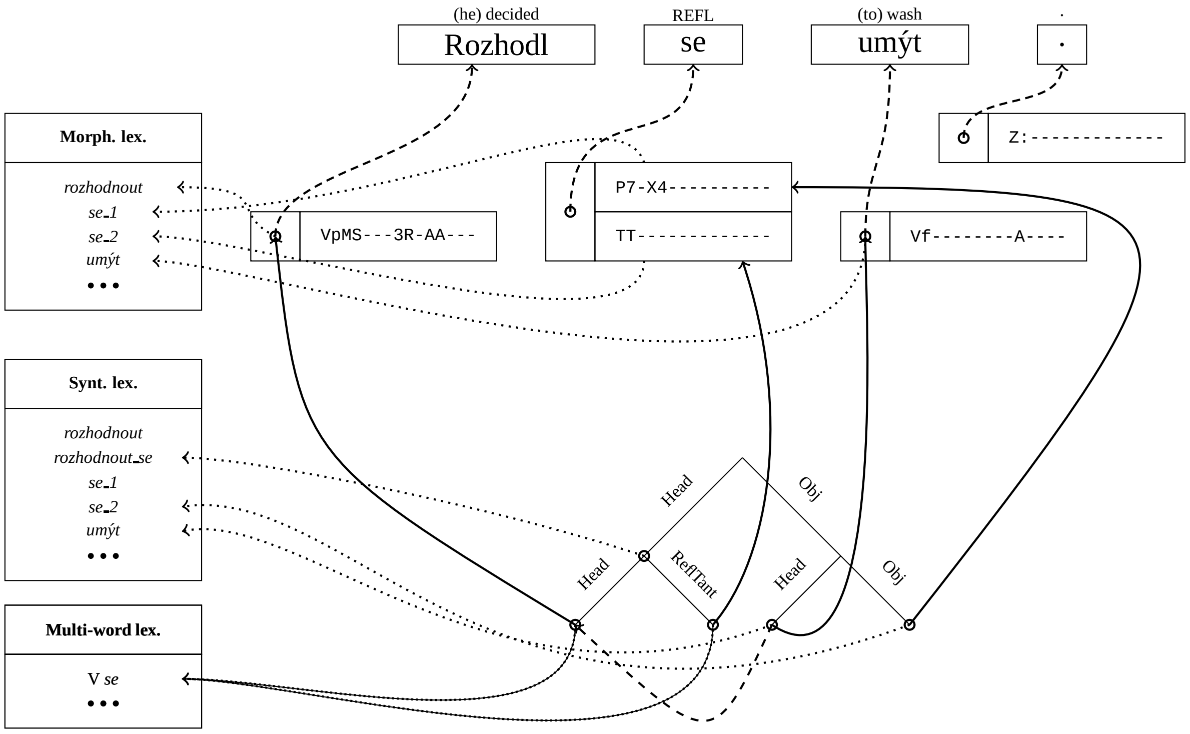

Figure 3: Solution to a case of haplology of a reflexive

whether in segmentation, morphology or syntax. Examples include valency slots with ambiguous case requirements filled by nouns exhibiting case syncretism as in (5) (Oliva, 2001), or structures involving PP-attachment ambiguity without a difference in meaning (4). Ambiguities of this type cannot be resolved even in a wide context.

(4)

Uzavřeli	mír	s	nepřítelem.
concluded	peace	with	enemy

‘They made peace with the enemy.’

(5)

V továrně	se	využívá	zařízení	na výrobu	kyslíku.
in factory	REFL	use	device$_{nom/acc}$	for production	oxygen

‘In the plant a device for the production of oxygen is used.’

Unresolved ambiguity may also be preferable to an arbitrary decision in case of poor evidence.

The scheme accommodates inherently ambiguous or undecidable phenomena using underspecification and distribu-

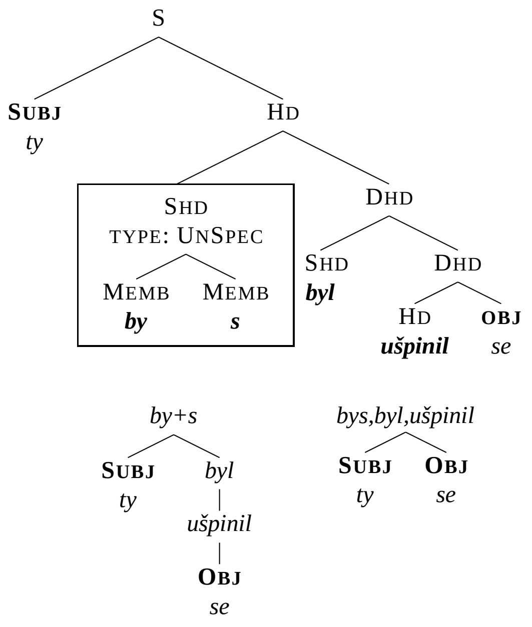

Figure 4: Three views of a single sentence

tive disjunction, both for category values and structures. Annotation of any kind can be missing; in the extreme case, syntactic structure of a sentence may consist of a mere list of words. A partial analysis may identify a word's head, its membership in a constituent, its syntactic function, or any combination of the above, while still leaving other syntactic relationships in the sentence unresolved.

To allow for such arbitrary underspecification, the skeleton structure is constituency-based, with a combination of binary and flat branching. Sub-constituents are specified by reference to a list of all constituents in sentence (6).[9]

(6)

Zdravotnictví	musí	zachránit	stát.
health service$_{nom/acc}$	must	save	state$_{nom/acc}$

#1 Health service must save the State.
#2 Health service must be saved by the government.

(7) Morphological analysis of (6) with some values unspecified:

$\boxed{1}$ zdravotnictví	*noun*, CASE=*X*, NUM=*sg*, GEND=*n*
$\boxed{2}$ musí	*verbfin*, PERS=*3*, NUM=*sg*
$\boxed{3}$ zachránit	*verbinf*
$\boxed{4}$ stát	*noun*, CASE=*Y*, NUM=*sg*, GEND=*m*

(8) Constituents in one of the two possible syntactic structures of (6), some boxed numbers refer to the forms above:

$\boxed{5}$ [$\boxed{3}$zachránit $\boxed{4}$stát]
$\boxed{6}$ [$\boxed{2}$musí $\boxed{5}$]
$\boxed{7}$ [$\boxed{1}$zdravotnictví $\boxed{6}$]

(9) Two possible structures with constraints on category values and overriding clauses:

#1 = $\boxed{7}$, *X=nom*, *Y=acc*

#2 = $\boxed{7}$, *X=acc*, *Y=nom*, $\boxed{1} \rightarrow \boxed{4}$, $\boxed{4} \rightarrow \boxed{1}$

[9]Note that the example is not inherently ambiguous – it has two distinct interpretations, potentially distinguishable given an appropriate context or world knowledge.

While dependency structure requires a specification of heads and dependency links for all parts of the tree, constituency structure allows for leaving the status of a constituent and relations within an embedded constituent unspecified. Constituency structure may be more useful even for representing ambiguities, at least when they can be rendered as underspecifications. E.g., it is not clear what kind of structure is correct in appositive structures such as (10).

(10)

vedoucí	katedry	profesor	doktor	Václav	Novák
head	department	professor	doctor	Václav	Novák

'Professor doctor Václav Novák, head of the department'

Ambiguities can either be present in the output of the parser, if it is run in an n-best mode, or they can be reconstructed by rules targeting typical cases. Moreover, PP-attachment ambiguities without semantic relevance are supposed to be tagged as such in the output of the parser, without generating multiple structures explicitly. For the time being, we intend to use the latter, somewhat unreliable, information wherever appropriate, and focus on experimenting with the reconstruction approach.

3. Converting dependency trees

Our syntactic trees are grown in a dependency-based nursery of McDonald's MST parser to the shape of the PDT a-level standard. The parser takes as its input a morphologically disambiguated sentence; the disambiguation is performed by a hybrid tagging system consisting of a rule-based component comprising about 2500 rules and a stochastic tagger. Syntactic trees produced by the parser are checked and rectified (see Section 4. below), and then converted to the internal annotation scheme and format, which differs from the input in the following aspects:

- In a different overall structure: the new scheme is based on constituency (phrase-structure) trees, e.g. with the subject a sister node to the clause's predicate.
- In a smaller set of syntactic functions (cf. Table 2).
- In a different account of word order, represented by links connecting unordered terminal nodes of the tree with their corresponding elements on the level of graphemics.
- In reference links, used, e.g., for connecting predicative elements (finite verb forms, infinitives, transgressives, nominal predicates) with their subject.

Rather than developing a full-fledged grammar of Czech from scratch and preparing appropriate training and testing data we used almost error-free training data in a format usable by the parser (customized for Czech) and also a vast amount of testing data in PDT (ca. 1.5 mil. tokens).

The conversion is performed by the application of a sequence of transforming rules to each input sentence. We demonstrate the conversion using (11) as an example.

(11)

Most,	který	byl	v	havarijním	stavu,	by	měl	sloužit
bridge,	which	was	in	disrepair	state,	would	have	to-serve

dalších	třicet	let.
further	thirty	years

‘The bridge, which was in a state of disrepair, should serve for a further thirty years.’

Sentence (11) is converted from the parser output (a-level of PDT) in Fig. 5 to the new format as in Fig. 6.

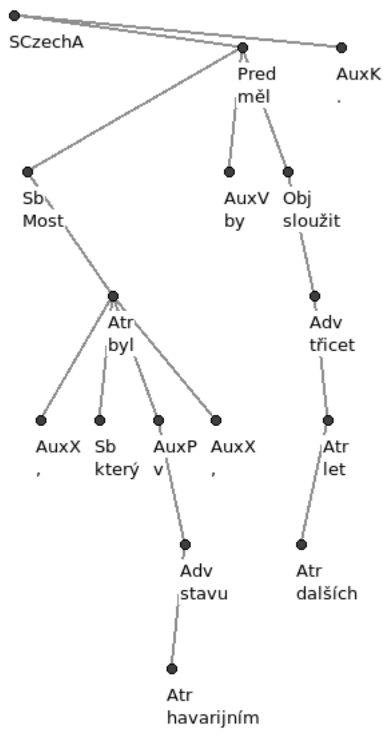

Figure 5: Syntactic tree in the PDT format

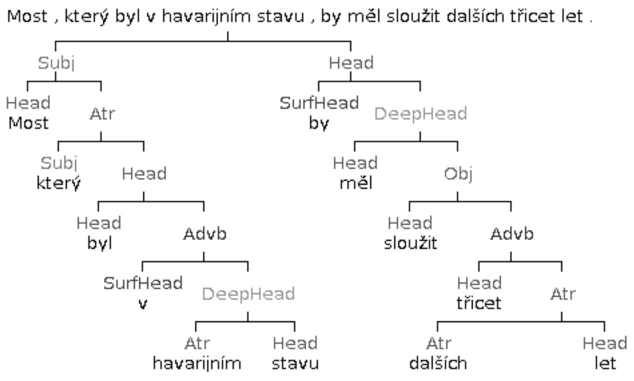

Figure 6: Syntactic tree in the new format

Some of the rules applied are merely technical or handle trivial operations on a single node. Other rules modify the geometry of the tree. Generally, the conversion of an input dependency tree to the corresponding constituency-based one proceeds recursively in a top-down direction, i.e., from the root of the input dependency tree down to the dependent nodes, and similarly, an output constituency-based tree is created. A rule is applied whenever the conversion algorithm finds the first appropriate bundle (= a simple tree).

In general, the input dependency bundle is converted in a following way: a new constituent is created having as its subconstituents a) all the constituents created by the transformation of the dependent nodes of the bundle and b) the constituent labeled by the output function HEAD or DEEPHEAD: this subconstituent will be assigned the word form, lemma and morphological properties of the head node of the dependency bundle.

A leaf of the input dependency tree is converted to the terminal constituent that is assigned the corresponding word form and its original function except for the nodes assigned the HEAD, DEEPHEAD or SURFHEAD function.

We shall briefly describe the main conversion rules, one of them a general rule, the others specific:

1. *General rule*. This rule converts a dependency bundle consistings of a head node and a non-empty set of its daughters to the corresponding constituent bundle. This bundle contains (i) the root and the nodes for the non-head daughters corresponding to the daughters of the input bundle and (ii) the node for the head daughter corresponding to the root of the input bundle: this node is assigned the HEAD function. The rule is applied to the input bundle whose head is a predicate and whose daughters are its non-subject complements.

2. *Subject rule and Auxiliary verb rule*. Both rules operate in a structurally identical way, the difference is in the syntactic functions. The *Subject rule* / *Auxiliary verb rule* take as input a bundle containing a root representing a verbal predicate and a *subject* / *auxiliary verb*, respectively, as one of the daughters of the root. On output, a binary constituent tree is created at the highest level: its head daughter corresponding to the verbal predicate is assigned the HEAD / DEEPHEAD function, the non-head daughter corresponds to the input *subject / auxiliary verb* and is assigned the SUBJ / SURFHEAD function. Furthermore, the head daughter is a root of another bundle: one of its daughters is assigned the HEAD function and the remaining non-head daughters correspond to the remaining sister nodes in the input.

3. *Prepositional group rule and Conjunction-clause rule*. These two rules also operate in a structurally identical way: they both transform an input dependency edge connecting a preposition / conjunction as the head element assigned the AuxP / AuxC function with the dependent node, referred to as X below. A binary tree is created on output: its root is assigned the function of the X node, the daughter corresponding to the preposition / conjunction is assigned the SURFHEAD function, the other one is assigned the DEEPHEAD function as well as the word form and morphological properties transferred from the input X node.

So far, 15 rules have been developed and tested on a subset of the PDT data. We plan to add new rules covering special phenomena encountered during intensive testing.

The conversion of the input structure of sentence (11) in Fig. 5 to the output structure in Fig. 6 can be briefly described as follows:

- First, the root of the input dependency tree labeled SCzechA as well as the node labeled AuxK representing a full-stop are left out, only the subtree with the root labeled by the Pred function will be converted.
- The Pred-labeled subtree will be processed in a left-to-right and top-down way, with every embedded bundle being recursively processed by the abovementioned rules. Thus, the Pred-labeled bundle with the three daughter nodes denoted Sb, AuxV and Obj will be processed first and subsequently the subtrees headed by them. The word forms and corresponding morphological tags (realized, in fact, by pointers to the morphological level and to lexicons, cf. Fig. 3 above) are recursively propagated down the constituency-based tree being created to be finally assigned to its leaf nodes.

In addition to the structure-changing rules used to generate phrase-structure trees complying with the new scheme, special rules adding reference links are applied.

Another group of rules, currently under development, are used to identify various substructures within the generated trees, such as:

- Agreement relations, such as subject – predicate, congruent attribute – noun, relative pronoun – antecedent
- Periphrastic verb forms including auxiliaries, such as conditionals, future and past tenses, passive
- Idioms and other specific types of collocations
- Inherently reflexive verbs or adjectives with the corresponding reflexive particles
- Surface/deep heads in structures of a specific type
- Non-projective (discontinuous) constructions (inferred from the surface order)
- Ambiguities undecidable even in a wider context (specific cases of PP-attachment and case syncretism)

Annotation of some of these structures (such as agreement relations and periphrastic forms) is not present in the treebank; the rules identifying them are invoked only after a user specifies his/her query to search for them in the treebank.

4. Automatic correction of results of the stochastic parser

Parsing unrestricted text by machine-learning techniques currently outperforms methods using hand-crafted rules, at least for Czech (Holan and Žabokrtský, 2006; Novák and Žabokrtský, 2007), therefore the input data for the conversion program are provided by stochastic parsers. Their accuracy up to now is, however, unsatisfactory: the best-performing parser available is the MST parser (McDonald et al., 2005) trained on PDT 2.5, whose success rate is 85.48% for unlabeled structures and 78.23% on labeled ones (measured on the *evaluation-test* subset of the PDT). A detailed analysis of the parser's results performed on the object corpus SYN2005 (100 mil. tokens of contemporary Czech) showed that it was possible to use linguistic knowledge for a reliable correction of many frequent parsing errors and thus to improve the overall accuracy of the parser. Therefore we developed our own rule-based correction tool with the set of rules being gradually enhanced.

During the development of this tool we annotated first the SYN2005 corpus by the stochastic parser. Then we manually checked the samples (ca 10,000 tokens altogether), in which we searched for recurrent errors, and subsequently we searched for errors in the whole corpus automatically. The most frequent errors were subject to further analysis and we tried to find reliable algorithms for error corrections. The parser makes both grave "grammatical" errors, where the resulting structure is totally inadmissible in the system of Czech syntax (e.g., two non-coordinated subjects depend on a single verb), and also common errors due to the parser's ignorance of valency, incorrect identification of head elements etc.

Automatic correction is performed by a modular software tool, employing extensive linguistic knowledge, represented, e.g., by various lists of lexical properties, including valency requirements. The tool identifies incorrect dependency structures, syntactic function labels and morphological tags. Whenever a well-known incorrect syntactic pattern is encountered, a corresponding correction routine is applied which chooses an appropriate correction according to the context and properties of the words in the erroneous structure. See (Jelínek, 2012) for more details.

One of the correction rules deals with subject incorrectly assigned to the noun in non-prepositional accusative. On the basis of context and properties of the incorrectly labeled word and its head verb the tool activates the appropriate linguistic rule which decides whether it should correct the syntactic function label (subject $\rightarrow$ object/adverbial), morphological tag (accusative $\rightarrow$ nominative in case of wrongly disambiguated case ambiguity) or the parent node in the dependency structure (some other appropriate head verb in the clause is selected).

So far we have created 30 correction rules, targeting more frequent errors for which reliable correction algorithms were found. One group of rules focuses on the basic dependency structure of a compound sentence consisting of verbs (they represent sentences in the PDT) and conjunctions. Another group of rules deals with incorrect syntactic functions and dependencies of syntactic nouns (e.g. the above-mentioned subject in accusative), the third group of rules rectifies the dependencies and functions of prepositional groups. The number of these rules grouped according their target domain and their success rate is shown in Table 3. The first column identifies the domain of the rules (dependency links and function labels of clauses, noun phrases, prepositional phrases and other rules), followed by the number of rules in the second column, the number of successful corrections of dependency links and syntactic function labels in the third and fourth columns, and the total number of successful corrections in the last column (some corrections modify dependency links as well as function labels). The success rate is calculated automatically on the *e-test* data set of PDT 2.5. The counts represent the number of successful corrections, decreased by the number of unsuccessful corrections, normalized per one million tokens. The last row shows the figures in percentages.

	rules	dependency	label	total
Clauses	6	1688	774	1744
NP	8	819	2066	2625
PP	11	834	7160	7722
Other	5	412	1390	1802
Total (ppm)		3753	11390	13893
Total (%)		0.38%	1.14%	1.39%

Table 3: Success rate of the correction modules

The resulting success rate improves from 85.48% unlabeled and 78.23% labeled accuracy to 85.86% unlabeled and 79.62% labeled accuracy. The correction tool can be further extended by additional rules.
In addition to extending the coverage of the correction tool, more procedures to improve the parsing results are in the pipeline, such as a rule-based pre-processing of both training and annotated data, a combination of multiple parsers,[10] a modification of the tagset to better serve the needs of parsing, and an extension of training data with text genres insufficiently covered in our original test set, such as fiction. We expect to raise the reliability of syntactic annotation above 87% unlabeled and 83% labeled by the end of the project.

5. Conclusion

We wish our treebank to match the size of POS-annotated corpora, while avoiding a theoretical bias by offering various views of syntactic annotation, based on a single core representation. The viability of this approach reflects the fact that linguistic theories share a broad common core. A sentence can then be visualized as a constituency-based or dependency-based structure with parameterizable underspecifications according to the user's wish. Three levels of representation (graphemic, morphological and syntactic) support the view of a bare input sentence and/or its morphological and syntactic annotation in various degrees of descriptive granularity. The system should satisfy demands of both an expert user and a student of syntax at higher elementary and secondary levels.
For a corpus of this size it would be unrealistic to count on manual checking of the output of automatic annotation tools. As a partial remedy, we use a rule-based correction module, targeting typical errors and inconsistencies. Together with visualisation options hiding very specific details or embedded structures, which a typical corpus user is expected to use as a preference, the effective error rate in the displayed data will be lower than in the output of the parser. We believe that the price for a significantly scaled-up treebank, paid in less reliable annotation, will be bearable for many tasks.

6. Acknowledgements

The authors are grateful to anonymous reviewers for very helpful comments. Work on this project was supported by the Grant Agency of the Czech Republic as project No. P406/10/0434.

[10]We plan to test and tune at least one additional parser (Nivre et al., 2006).

7. References

Jan Hajič. 2006. Complex Corpus Annotation: The Prague Dependency Treebank. In Mária Šimková, editor, *Insight into the Slovak and Czech Corpus Linguistics*, pages 54–73, Bratislava, Slovakia. Veda.

Annette Hautli, Sebastian Sulger, and Miriam Butt. 2012. Adding an annotation layer to the hindi/urdu treebank. *Linguistic Issues in Language Technology – LiLT*, 7, January.

Tomáš Holan and Zdeněk Žabokrtský. 2006. Combining Czech dependency parsers. In Petr Sojka, Ivan Kopeček, and Karel Pala, editors, *TSD*, volume 4188 of *Lecture Notes in Computer Science*, pages 95–102. Springer.

Tomáš Jelínek. 2012. Automatic rule-based correction of stochastic syntactic annotation of Czech. In Markéta Ziková and Mojmír Dočekal, editors, *Slavic Languages in Formal Grammar. Proceedings of FDSL 8.5, Brno 2010*. Peter Lang, Frankfurt am Main.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 523–530, Stroudsburg, PA, USA. Association for Computational Linguistics.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC-2006*, pages 2216–2219, Genova. ELRA.

Václav Novák and Zdeněk Žabokrtský. 2007. Feature engineering in maximum spanning tree dependency parser. In Václav Matoušek and Pavel Mautner, editors, *Lecture Notes in Artificial Intelligence, Proceedings of the 10th International Conference on Text, Speech and Dialogue*, volume 4629 of *Lecture Notes in Computer Science*, pages 92–98. Springer, Berlin / Heidelberg.

Karel Oliva. 2001. On retaining ambiguity in disambiguated corpora. *TAL (Traitement Automatique des Langues)*, 42(2).

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Adam Przepiórkowski and Alexandr Rosen. 2005. Czech and Polish raising/control with or without structure sharing. *Research in Language*, 3:33–66.

Adam Przepiórkowski. 2007. On heads and coordination in valence acquisition. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing (CICLing 2007)*, Lecture Notes in Computer Science, pages 50–61, Berlin. Springer-Verlag.