Towards a treebank for all tastes

Petr Jäger, Vladimír Petkevič,
Alexandr Rosen and Hana Skoumalová
(Charles University, Prague)

1. Introduction

Syntax is a discipline of many theories, and it is accordingly difficult to build a syntactically annotated corpus that would not put off at least some syntacticians by an alleged or real theoretical bias. Yet despite appearances and focus on slightly different sets of linguistic phenomena the theories strive to describe and explain the same object – a natural language. In fact there is a large pool of implicit wisdom shared by all syntactic theories and a significant overlap of linguistic knowledge can be extracted from all theory-specific formats. Thus a treebank offering different views of syntactic annotation while based on a single core pattern need not be a dream out of touch with reality. In addition to constituency and dependency trees of various shapes, suited to the taste of experts in linguistics, one of the views may be close to the representation of syntactic structure to which Czech students are exposed at the higher elementary and secondary levels.

Such a treebank should indeed be useful beyond academic community to other professionals and lay users interested in language and linguistics. Obviously, for most of them the bigger the better, but not at an unbearable decrease in reliability. Yet the largest existing treebanks reach the relatively modest sizes of several million words, an insufficient number for many tasks. The reason is the cost of manual checking needed to improve the error rate of automatic syntactic annotation tools, which still perform much less reliably than part-of-speech taggers. However, to match the size of a balanced POS-tagged corpus, the use of automatic parsing tools without manual checking is inevitable.

Building on previous efforts in treebank annotation, especially the Prague Dependency Treebank (PDT) and the NEGRA/TIGER Corpus (Hajič, 2006; Hajič et al., 1998; Skut et al., 1997, i.a.) we want to make a further step towards a large corpus with a reasonably reliable, automatically assigned syntactic annotation. With this aim in mind, we propose an explicitly defined annotation scheme consisting of a linguistically founded, 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, accessible to lay users and satisfying demands of experts at the same time (§2).

. Work on this project was supported by grant GAČR P406/10/0434.

Our claim is that a large and reasonably reliable treebank can be built using a stochastic parser (Holan & Žabokrtský, 2006), a rule-based correction module, diminishing the parser's error rate (see §5 and Jelínek, 2011), and customizable visualization options, potentially less sensitive to errors in details or more embedded constituents.
The proposed annotation scheme should be useful even in a different context, where phenomena difficult to handle by automatic methods are annotated manually in a smaller treebank. Rather than tailoring our scheme to suit the possibilities of available tools, we prefer to reflect potential requirements of the corpus user and base the scheme on concepts open to the options of representing inherent ambiguities (impossible to resolve even in a wider context), pronominal references and other phenomena that may require some manual effort.

2. The annotation scheme

Key features of the annotation scheme are listed together with reasons for their introduction and brief hints on how the relevant information can be gained (see §4 for more details).

2.1 Multiple options to display syntactic structure

While presenting an easy, friendly interface to the lay user, the syntactic annotation scheme does not impose a single way of representing syntactic structure. To offer different views of syntactic structure, the core representation can be interpreted as 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 with an arbitrary amount of detail, not necessarily by tree graphs. A linear display identifying the major (possibly discontinuous) constituents of a clause by different colors or typefaces could be the option of choice for many users, see (1).

(1) A linear display of elementary syntactic structure:

Ty by *se*s BYL UŠPINIL.

An important side effect of less detailed visualization is that some annotation errors can remain hidden.

2.2 Ambiguity and partial information

Corpus annotation is mostly unambiguous. Yet ambiguity is sometimes inevitable for fundamental reasons, whether in segmentation, morphology or syntax. Examples include valency slots with ambiguous case requirements filled by nouns exhibiting case syncretism as in (2) (Oliva, 2001), or (quite common) structures involving PP-attachment ambiguity without a difference in meaning. Ambiguities of this type cannot be resolved even in a wide context or with extensive world knowledge.

(2) V továrně se využívá zařízení na výrobu kyslíku.
in factory REFL use device$_{gen/acc}$ for production oxygen
'In the plant a device for the production of oxygen is used.'

(3) Uzavřeli mír s nepřítelem.
concluded peace with enemy
'They made peace with the enemy.'

Additionally, unresolved ambiguity may be preferable to an arbitrary decision in case of poor evidence or some other insufficiency.
The scheme accommodates inherently ambiguous or undecidable phenomena using underspecification and distributive 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. Unresolved ambiguity is not our preferred solution if unambiguous interpretation is attainable, but we wish to leave it as an option for all other cases.
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 (4).[1]

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

The intended meaning of the text attributes is as follows: **subject**, predicate, *object*, AGREEING FORMS. Highlighting by different colors is not used for typographical reasons.

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

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

(5) Morphological analysis of (4) with some values unspecified:
[1] zdravotnictví *noun*, CASE=*X*, NUM=*sg*, GEND=*n*
[2] musí *verbfin*, PERS=*3*, NUM=*sg*
[3] zachránit *verbinf*
[4] stát *noun*, CASE=*Y*, NUM=*sg*, GEND=*m*

(6) Constituents in one of the two possible syntactic structures of (4), some boxed numbers refer to the forms above:
[5] [[3] zachránit [4] stát]
[6] [[2] musí [5]]
[7] [[1] zdravotnictví [6]]

(7) Two possible structures with constraints on category values and overriding clauses:
#1 = [7], *X=nom*, *Y=acc*
#2 = [7], *X=acc*, *Y=nom*, [1] → [4], [4] → [1]

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.

2.3 Surface and deep structure

Every constituent of the new scheme is either of the headed or unheaded type and is also assigned a syntactic function. The whole repertory of types and functions is presented in Tables 1 and 2 below.

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
OBJ_ADVB	Object or Adverbial with two subtypes:
– OBJ	– Object
– ADVB	– Adverbial
ATTR	Attribute
VBATTR	Verbal complement
REFLTANT	Reflexive particle
DEAGENT	Reflexive particle with the deagentive meaning
APOS	Apposition
INDEP	Independent constituent (parenthesis, noun in the vocative, etc.)
MEMB	Syntactic daughter of an unheaded constituent

Table 2: Syntactic functions

As Table 2 shows, a head can be distinguished as surface or deep; a function word such as preposition or verbal auxiliary is labelled as surface head while its sister is the deep head.[2] This allows for extracting both surface and deep dependencies from a single structure, see (9). Coordination and similar constructions are treated as headless (they are of the type UNHEADED).

(8) Ty by ses byl ušpinil.
You would REFL+AUX*2nd,sg* be*pple* get dirty*pple*
'You would have got dirty.'

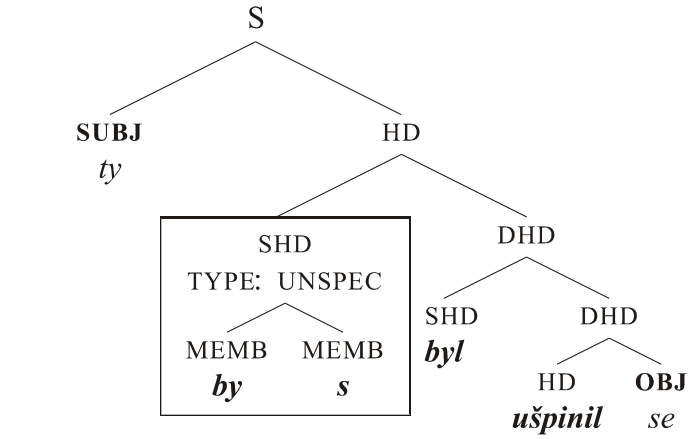
The three structures in (9) are all possible renderings of a single analysis of (8). The constituent structure has function labels for subject, object, head, surface head and deep head, and it is followed by the derived surface and deep dependency structures.[3] Complex verb forms are highlighted by boldface,

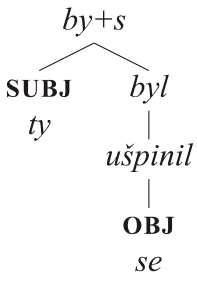
2 Przepiórkowski (2007) distinguishes syntactic and semantic heads.

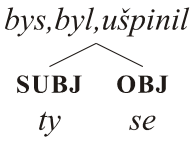
3 For technical reasons, the labels mark nodes rather than edges, representing both constituency and functional relations. The nodes refer to categorial information

contractions by a box.

(9)







appropriate to words or phrases, as in the analysis of (4) above.

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 labelled 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.

2.4 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 visualization 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 (10), and contractions such as *ses*, represented as a single graphemic unit, are analyzed as two morphological forms: here as a reflexive pronoun/particle and a 2nd person auxiliary. More mismatches in the number of tokens occur between the levels of morphemics and syntax, where punctuation is omitted.

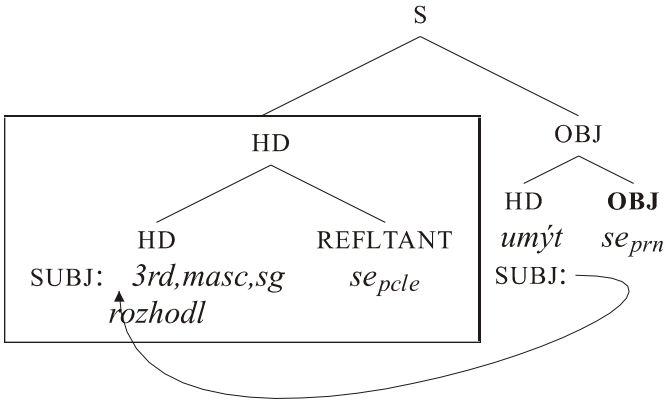
(10) Rozhodl se umýt.
Decided$_{masc,sg}$ REFL wash$_{inf}$
'He has decided to wash himself.'

The haplologized item *se* is both a reflexive particle, a part of the inherently reflexive verb *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*	*prnre,acc*	*inf*

(11)

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.



(12)

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 (12), PRO is represented equivalently as a link. All such phenomena are represented by linking the infinitive, predicative complement, base coordinated verb etc. across the structure with its argument. The link is labelled by the relevant syntactic function, see (12).

Other links 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 (1), agreeing forms are shown in capital letters. Depending on the user's choice, discontinuous (non-projective) structures can be represented as such, with crossing branches in 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 identifies non-projectivity in the assumed dependency structures, and its results will be subject to checks and modifications by the correction rules also in this pocket of syntax. The conversion module spots additional discontinuities which only occur in the phrase structure.

2.5 Lexicon and grammar

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 is used to index word tokens using lemmas with appropriate categories, as well as compound forms and multi-word lexical units.

3. Encoding the annotation

There are multiple options for the encoding of treebanks, and deciding about the proper choice is not easy. For a recent overview and evaluation of existing standards and implementations see Przepiórkowski & Bański (2009) and Bański & Przepiórkowski (2010).

The most straightforward option for us seemed to stick to the format of our primary source of linguistic data, the output of the stochastic parser. This is the data format developed for the TectoMT suite, which includes the parser we use.[4] The format is built on top of the XML-based Prague Markup Language, used mainly to encode the multi-level annotation in the Prague Dependency Treebank,[5] but it is capable of representing constituency-based trees and can be adapted for various other tasks. However, it does not lend itself easily to some design goals for our annotation scheme, such as the distinction between surface and deep heads and those related to representing underspecification and ambiguities, including the level of tokenized text, the option of unstructured constituents and other variant representations, interlinked across levels.

Among the available standards, the Text Encoding Initiative guidelines.[6] seemed

4 See http://ufal.mff.cuni.cz/tectomt/.

5 See http://ufal.mff.cuni.cz/jazz/PML/.

6 See http://www.tei-c.org/.

to be the most promising contender, but in the end we arrived at the conclusion that there is not much benefit in picking and choosing from a pool of recommended options, while having to design solutions to issues that do not seem to have a natural implementation in the standard.

Our purpose-designed format reflects the annotation scheme by introducing three levels: graphemics, morphology and syntax. The level of graphemics consists of tokens (minimal text strings), stand-off annotated by the higher levels. The level of morphology consists of morphologically annotated words. A word may consist of multiple tokens (for frozen sequences without structure), or a single token may be decomposed into several words (for contractions). A single string of tokens may be interpreted in more than one way. Variant sequences of words make sure that words in one reading of the strings do not overlap.

A schematic picture is shown in Fig. 1, followed by a sample of XML encoding in Fig. 2. The sentence, consisting of two tokens, is two-way ambiguous both at the level of morphology and syntax (see (13) and (14)).

(13) Ohlas to.
Bend$_{l\text{-}pple,fem,sg}$+AUX$_{fin,2nd,sg}$ it
'You have bent it.'

(14) Ohlas to.
Report$_{imperative,2nd,sg}$ it
'Report that.'

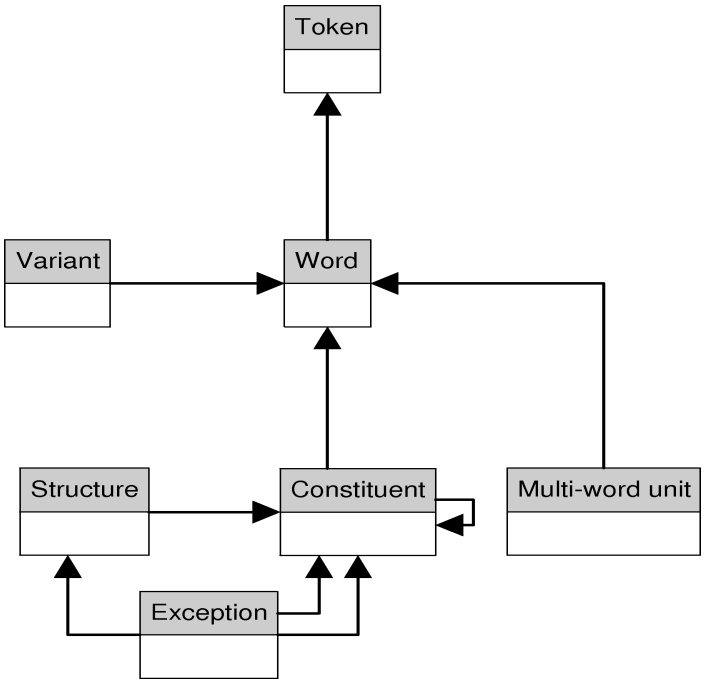


Figure 1: Overview

The level of syntax consists of constituents, labelled by functions. As in morphology, an ambiguous sequence may be interpreted in more than one way. The structure element provides a reference to the top constituent(s). To avoid proliferation of structures in the case of multiple local ambiguities, an embedded constituent may have one or more alternatives with a different internal setup,

leaving the rest of the structure unaffected.

```
<?xml version="1.0" encoding="utf-8"?>
<sentence>
  <head />
  <graphemics>
    <token order="1000" value="ohlas" type="word"/>
    <token order="2000" value="to" type="word"/>
  </graphemics>
  <morphology>
    <word id="5" order="500" lemma="??"/>
    <word id="1" order="1000" wcl="verb.imper" pers="2" num="sg" lemma="ohlásit" />
    <word id="2" order="1000" wcl="verb.lpple" gend="fem" num="sg"
     lemma="ohnout" />
    <word id="3" order="1010" wcl="verb.fin" pers="2" num="sg" lemma="být">
      <token_ref order="1000"/>
    </word>
    <word id="4" order="2000" wcl="ppron" case="acc" num="sg" gend="neut"
     lemma="to" />
    <variant id="1">
      <word_ref id="1" />
      <word_ref id="4" />
    </variant>
    <variant id="2">
      <word_ref id="5" />
      <word_ref id="2" />
      <word_ref id="3" />
      <word_ref id="4" />
    </variant>
  </morphology>
  <syntax>
    <constituent id="1">
      <word_ref id="1" function="head" />
      <word_ref id="4" function="obj" />
    </constituent>
    <constituent id="2">
      <word_ref id="5" function="sb" />
      <constituent_ref id="3" function="head" />
    </constituent>
    <constituent id="3">
      <word_ref id="3" function="shead" />
      <constituent_ref id="4" function="dhead" />
    </constituent>
    <constituent id="4">
      <word_ref id="2" function="head" />
      <word_ref id="4" function="obj" />
    </constituent>
    <structure id="1" constituent_ref="1" rating="1000" />
    <structure id="2" constituent_ref="2" rating="1000" />
  </syntax>
</sentence>
```

Figure 2: XML encoding of a sample ambiguous sentense *ohlas to*

Some concepts cut across the basic constituency structure. Links may be used to represent pronominal references and agreement. An important part of the format is the concept of multi-word units, used to identify analytical verb forms, potentially discontinuous multi-word lexical items and phrasemes.

4. Converting dependency trees

Our syntactic trees, grown in a dependency-based nursery of McDonald's MST parser to the shape of the PDT a-level standard, are checked and rectified (see §5 below), and then converted to the internal annotation scheme and format, which differs from the input in the following aspects:

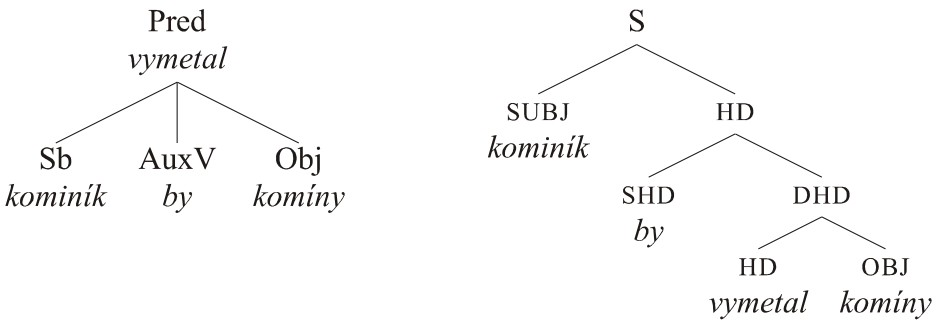
- In a different overall structure: the new scheme is based on constituency (phrase-structure) trees, e.g. with the subject a sister node to the clause's predicate.
- In a smaller repertory of syntactic functions.
- In a different account of word order, represented by links connecting unordered terminal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

The conversion is performed by the application of a sequence of transforming rules to each input sentence. We show the process of conversion using (15) as an example.

(15) Kominík by vymetal komíny.
chimney-sweep would sweep chimneys
'The chimney-sweep would sweep the chimneys.'

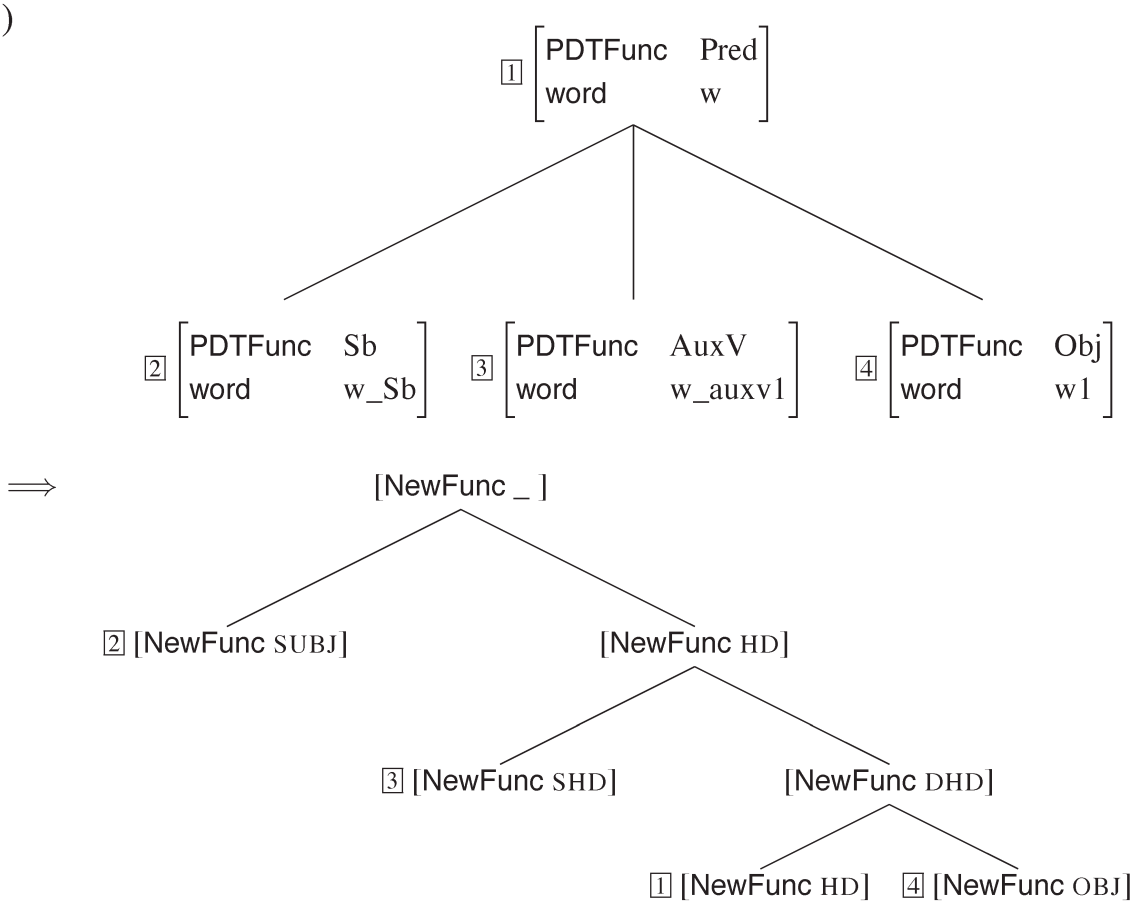
Sentence (15) is converted from the parser output (a-level of the PDT standard) to the new format as in (16):

(16)



The input is subject to the application of a sequence of rules, some of which are merely technical or handle trivial operations on a single node. Other rules, such as that in (17), modify the geometry of the tree. This rule converts the dependency structure with the governing predicate (Pred – *vymetal*) and its dependent nodes for subject (Sb – *kominík*), auxiliary verb (AuxV – conditional particle *by*) and object (Obj – *komíny*) to the corresponding constituent structure. The predicate part labelled HD has two daughter nodes: for the conditional by (SHD) and the rest of the predicate part (DHD). This node has two daughter nodes for the content verb (HD – *vymetal*) and its object (OBJ – *komíny*).

(17)



In addition to the structure-changing rules (numbering 20 at most), special rules adding reference links are applied (no such rule was necessary in our example). The rules above are used to generate phrase-structure trees complying with the

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

- Agreement relations of various types, 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 and deep heads, constituting structures of a specific type
- Non-projective (discontinuous) constructions (inferred from the surface order)
- Ambiguities undecidable even in 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.

5. Improving on the output of a stochastic parser

Parsing unrestricted text by machine-learning techniques currently outperforms methods using hand-crafted rules at least in coverage, although their error rate may still be too high. A way to a reasonably reliable syntactic annotation seems to be a combination of linguistic and stochastic methods.
The overall accuracy of the annotation is improved by applying linguistically motivated rules to the output of a McDonald's MST Parser (Holan & Žabokrtský, 2006), a tool included in the TectoMT package (Žabokrtský et al., 2008). In a specific combination with other taggers, its success rate of 86% makes it currently the best performing parser of Czech.[7] The output consists of dependency trees, corresponding to the levels of surface and underlying syntax (*a-level, t-level*) of the Prague Dependency Treebank.[8] Syntactic structure, syntactic functions and other relevant information identified by the parser are

7 The parser's success rate may drop by up to 2 for some type of texts.

8 Currently, only the a-level is used, but both syntactic levels of PDT will be useful: only *t-level* includes explicit referential links.

extracted from the PDT format and transformed into the new annotation scheme. The parser's performance is being evaluated in terms of recurrent error types in a test corpus. Based on this evaluation, linguistically motivated rules are designed and applied to the parser's output (see Jelínek, 2011 for more details). So far, the rules operate on the dependency-based structures in the source format of the TectoMT package, but more rules will be used later within the target format, whenever the source format lacks expressive power. At present, these rules improve the result by approx. 7%, increasing the overall success rate in the ideal case from 86% to 87%.

6. 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 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 visualization 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.
In order to achieve the best possible results, we will focus on optimizing the rule-based correction module and on tuning the performance of the whole setup of the automatic annotation tools.

References

Bański, Piotr & Adam Przepiórkowski, 2010. The TEI and the NCP: the model and its application. In *LREC 2010 Workshop on Language Resources: From Storyboard to Sustainability and LR Lifecycle Management*. Valletta, Malta: ELRA.

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

Hajič, Jan, Eva Hajičová, Jarmila Panevová & Petr Sgall, 1998. Syntax v Českém národním korpusu. *Slovo a slovesnost*. 59, 168–177.

Holan, Tomáš & Zdeněk Žabokrtský, 2006. Combining Czech Dependency Parsers. In *Proceedings of the 9th International Conference on Text, Speech and Dialogue*, Berlin/Heidelberg: Springer. 95–102.

Jelínek, Tomáš, 2011. Automatic rule-based correction of stochastic syntactic annotation of Czech. In *this volume*.

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

Przepiórkowski, Adam & Piotr Bański, 2009. Which XML standards for multilevel corpus annotation? In *Proceedings of the 4th Language & Technology Conference*. Poznań, Poland, 245–250.

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

Skut, Wojciech, Brigitte Krenn, Thorsten Brants & Hans Uszkoreit, 1997. An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing ANLP-97*. Washington, DC.

Žabokrtský, Zdeněk, Jan Ptáček & Petr Pajas, 2008. TectoMT: Highly Modular MT System with Tectogrammatics Used as Transfer Layer. In *Proceedings of the 3rd Workshop on Statistical Machine Translation.* Columbus, Ohio: ACL. 167–170.