Post-annotation Checking of Prague Dependency Treebank 2.0 Data

Jan Štěpánek

Abstract

Various methods and tools used for the post-annotation checking of Prague Dependency Treebank 2.0 data are being described in this article. The annotation process of the treebank was complicated by several factors: for example, the corpus was divided into several layers that must reflect each other. Moreover, the annotation rules changed and evolved during the annotation. In addition, some parts of the data were annotated separately and in parallel and had to be merged with the data later. The conversion of the data from an old format to a new one was another source of possible problems besides omnipresent human inadvertence. The checking procedures used to ensure data integrity and correctness are classified according to several aspects, e.g. their linguistic relevance and their role in the checking process, and prominent examples are given. In the last part of the article, the methods are compared and scored.

1 Introduction

The annotation of a corpus is always a complex task. In the case of Prague Dependency Treebank 2.0 (PDT) (Hajič et al., in preparation), the situation was even more complicated: the corpus contains not only a morphological annotation (linear), but also a syntactic (i. e. structural) one which is much more complex than a simple linear annotation. The main source of errors is the human lack of concentration, but there are other factors: the annotation is divided into several layers that are interlinked by references according to stand-off annotation principles, thus a change at one layer may produce an error at a different one, for which the annotator does not have to be specialised. Moreover, annotation rules changed and evolved during the annotation process, possibly turning valid data into invalid ones, as they were not re-annotated every time a rule changed.[1]

Other possible sources of errors were conversions from old one-purpose formats to the new XML-based format (Pajas and Štěpánek, 2005), merging the data with parallel annotation (topic-focus articulation, coreference, valency frames) and various automated procedures.

As the data were being collected, revised or used for further research, as in (Ondruška, Panevová, and Štěpánek, 2003), many errors of linguistic nature were found. The tools originally used just for basic checking of the data validity showed to be suitable for deeper corrections based on linguistically motivated invariants.

2 Prague Dependency Treebank

The PDT is based on the theory of Functional Generative Description, as proposed in (Sgall, 1967) and later elaborated in (Sgall, Hajičová, and Panevová, 1986). Linguistic description of a sentence according to the theory consists in its representation at several layers, from the phonetical layer, through the morphonological, morphemic and surface syntax one to the tectogrammatical (deep syntax) layer. Items from neighbouring layers are connected by a relation of representation that corresponds to the notion of *function and form*: for example, the subject (item of the surface syntax layer) has usually the form of

[1] Problems with annotation of the previous version, Prague Dependency Treebank 1.0 (Hajič et al., 2001), was quite similar. Only a few checking procedures were used in that case, though.

the nominative case (morphematical), whilst the function of the nominative case is generally the subject. The syntax layers both use dependency structures (trees) to represent relation between words in a sentence.

PDT data are divided into four layers. The "lowest" layer (w) contains the original source text divided into documents and tokens. The "core" three layers, morphological (m), analytical (a) and tectogrammatical (t), contain human annotation. During the annotation process lasting from 1996 till now, the volume of the annotation has grown to 2 million words at the m-layer, of which 1.5 million were annotated at the a-layer as well. 0.8 million of them were annotated at the t-layer.

The m-layer (Zeman et al., 2005) segments text into sentences and assigns a *lemma* and *tag* to every token of the w-layer. Lemma corresponds to the noninflected basic form of a word and tag contains information about all the relevant morphological categories. On the m-layer, corrections of mistakes and typos from the w-layer were included (see Subsection 4.1).

The a-layer (Hajič, 1998) contains the surface syntactic structure. Every token of the m-layer corresponds to one node of a dependency tree (and vice versa, each node corresponds to one m-layer token). The type of a tree edge is indicated by the *analytical function* assigned to the dependent node (i.e. the one farther from the root).

The t-layer encompasses the deep syntactic structure and many additional features. Some nodes in dependency trees at the t-layer do not correspond to any surface tokens (they represent dropped subject or ellipsis), some tokens have no corresponding node in the t-tree (e.g. punctuation); sometimes, several surface tokens correspond to one t-node (e.g. a preposition and noun or a conjunction and verb and modal and auxiliary verbs). The situation in which one surface token corresponds to several t-nodes is also possible (e.g. word elided not to be repeated on the surface, as in *"Peter gave Mary a flower and* [gave] *John a book"*).[2] The type of a tree edge is indicated by the *functor* (and *subfunctor* in some cases).

The additional features include the following:

- *Grammatemes* are attached to some nodes at the t-layer, providing information not derivable from the structure (e.g. modality and tense for verbs).
- Every node representing a verb (and some nodes representing adjectives or nouns) is assigned a *valency frame*.
- Each node is assigned one of the three values assigned on the basis of contextual boundedness or *topic–focus articulation*: a node can be contextually bound, contrastively contextually bound, or contextually non-bound. In addition, the nodes in the topic part of the sentence are ordered according to the assumed communicative dynamism.
- At the current phase of annotation, *coreference* relations between nodes of certain category types are captured, distinguishing also the type of the relation (textual, grammatical, or the "second dependency" of complement).

Several more features are mentioned in Subsection 4.4.

3 Tools

The main tool used for the PDT data processing was Tree Editor TrEd (Hajič, Vidová-Hladká, and Pajas, 2001). TrEd is written in the scripting language Perl and has several advantages to other and previously used tools:

[2] Even the $m : n$ correspondence is possible, e.g. if a word with a preposition was elided (*"he agreed with the extension of the territory over the Western part and later* [he agreed **with the extension**] *over the Southern one"*).

- Since it is written in Perl, TrEd can be run under diverse operating systems (MS Windows, Linux etc.).
- Perl is an interpreted language. Therefore, Perl itself can be used to write "macros" — pieces of code that use the libraries for data processing that can be run from within the editor.
- TrEd is open-source and well documented, which means good accessibility and flexibility.

Besides the tree editor, there exists a "console" version of the data processor, called btred. It can search and/or change the data with all the benefits of TrEd, but without calling the graphical user interface. However, the corpus is too large to be processed by it, one file by one: it takes hours to traverse all the files, which makes debugging almost impossible. For that purpose, the distributed version ntred was developed and became the main tool for the post-annotation checking.

Many minor tools were used in the checking process as well, mainly scripts in Perl and bash.

4 Classification of Checking Procedures

The checking procedures (CP's) can be classified according to several aspects:

1. **Origin of the procedure.** Most of the checking procedures were inspired by the annotation guidelines. Some arose at a randomly found annotation error to detect whether the error occurred more than once. Many were inspired by classifying the output of the procedure that was searching for identical surface strings with differing annotations.
2. **Purpose of the procedure.** The procedures were divided into four sets.
   - The first set, called "find", contained 469 procedures that just searched for suspicious data, i. e. positions with probable occurrences of sought errors. The output of such a procedure was processed either by a procedure from the second set, if it was possible to handle the problem without human assistance, or by a human annotator.
   - The second set, called "fix", comprehended 135 procedures that were able to change the data and repair errors.
   - The third set, called "check", contained 196 procedures similar to those in the `find` set, but they included lists of exceptions for rare cases where the supposed invariant did not hold. All procedures from the `check` set were expected to report no errors on the whole data.
   - The last set, called "misc", contained a variety of procedures that behaved in a different way than those in the first three sets.
3. **Layer of the data concerned.** The w-layer contains just the original source text and hence needed almost no CP's. Remaining three layers contain human annotation and were subject to CP's. An additional group of CP's dealt with the data format and other low-level features of the treebank data that could be possibly broken.[3]

In Subsections 4.1, 4.2, 4.3, and 4.4, examples of checked invariants are given, assorted by the layer they were expected to correct.

4.1 Low-Level Checking Procedures

Low-level CP's have no linguistic relevance; they concern the data format, annotation scheme, or data representation. They condition any other, more sophisticated queries, and for that they had the highest priority.

[3] Some of the CP's concerned more than one layer, of course. They have been classified according to the most suitable group.

Number of files: Number of files had to be the same all the time. Lower number typically indicated an accidentally deleted file.

File format: All the files had to be loadable by btred. This procedure was used to detect corrupted files.

Attribute values: Some of the attributes have only limited sets of possible values. For some attributes, the sets changed during the annotation as some new values were added or some values were merged.

Uniqueness of identifiers: Identifiers had to be unique over the whole data. This procedure was complicated by the cluster architecture of ntred, because even if the identifiers had been unique at each server, there could have still been duplicities among different servers.

Reference validity: All the references were expressed as links to identifiers. This procedure verified that all referenced units really existed. As in the previous case, the cluster architecture made the procedure harder.

Linguistic and technical root: On the a-layer and t-layer, each sentence was represented by a dependency tree. The tree has a "technical" root that does not correspond to any sentence member but rather contains information about the sentence itself. The root must be the first node in the node ordering — otherwise the tree is suspect to be corrupted.

On the t-layer, the only child of a technical root has to be a "linguistic" root corresponding to the main word (most often the predicate) of the sentence. More children originated either in annotation error or in wrong sentence segmentation (see below).

Hidden added nodes: "Added" nodes were created by the annotators on the t-layer, typically for omitted valency participants or ellipsis. "Hidden" nodes, on the other hand, corresponded to surface auxiliary words that were not represented as nodes on the t-layer. While hiding a node, all its children were hidden, too. That way added nodes might get accidentally hidden.

Inter-layer linking: There were 35 procedures in the `find` set dealing with inter-layer links. For illustration, note that in the t-representation of the sentence *"They know that she had to go home"*, there is only one node *"go"* corresponding to all the a-nodes *"that", "had", "to"*, and *"go"*. Not only modal verbs and conjunctions were handled this way, but also prepositions and other auxiliary words.

In the released data there exists no a-node with no link from the t-layer (with some exceptions as punctuation etc.).

Some errors in the inter-layer linking were caused by the "copying" process on the t-layer: if a word was missing in the sentence, but it was expressed in the previous context, the annotator could "copy" the node so it was no longer missing (cf. *"Peter bought flowers and Andrew cakes."* — the node for *"buy"* would be copied). Not all the links to lower layers from the original node should be preserved at the copied node, though. For example, in sentences *"A reader might think that we are not talking about a serious problem. About a problem that concerns everyone."*, the word *"talking"* is copied to the second sentence, but without the auxiliary conjunction *"that"*.

Wrong sentence segmentation: The source text was segmented by a program that might make some errors. Several heuristics were used to determine such cases: a preposition without a word in the corresponding case, a sentence ending with a comma or other non-typical character, or a sentence starting with a lowercase letter.

Repairing such errors was very complicated, since in many cases it was not possible to simply split or join the sentences somewhere: continuous segment at the a-layer does not always correspond to a continuous segment at the t-layer and vice versa. Some nodes had to be moved from one sentence to another separately and the sentence had to be re-annotated at both the layers.

Word segmentation: Problems of word segmentation were similar to the problems of sentence segmentation. For example, a number representing date (*"6. 11."*) could be tokenised as a number with decimal point (*"6.11"*) followed by a dot. Therefore, all numbers with a decimal point that could be transformed to dates had to be checked manually.

Reason of form change: All the original forms of words are kept at the w-layer. In case of a misspelled word or normalisation of a numeral, the changed or normalised form was recorded into the m-layer together with the reason of the change. Sometimes, no reason was given, but the form had been changed.

4.2 Morphological Checking Procedures

Imperatives and vocatives: Most word forms interpretable as imperatives or vocatives were annotated misleadingly with that interpretation.

Local case without a preposition: In Czech, every word in local case must have a preposition. This CP could take advantage of both the a-layer and t-layer.

Negation: Capturing of negation was not coherent on the m-layer, which later caused problems at the t-layer where negation had to be represented as a separate node.

Agreement of adjectival attribute: Attribute in adjective has to agree in case, gender and number with its parent. The a-layer could be used to find attributes.

Rule based error detection: All the data were tested by the rule based disambiguation program (Květoň, 2006). Marked errors were corrected manually.

4.3 Analytical Checking Procedures

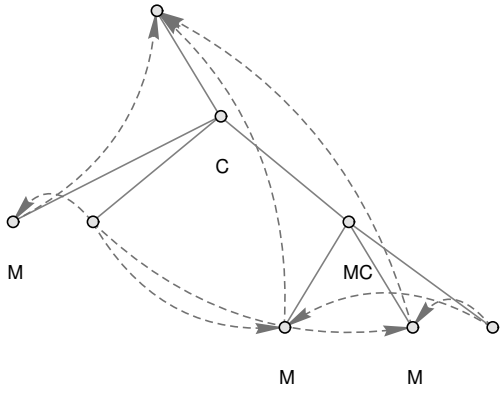

Figure 1: Coordination construction scheme.

Coordination: Coordination and apposition (and mathematical operations on the t-layer) were annotated in a special way: the coordinating conjunction is captured as a child of the node on which all the coordinated words depend. The words become children of the conjunction node with a special attribute `is_member`, while common dependent nodes are captured as children of the conjunction without the member attribute (see Figure 1: C denotes conjunctions, M denotes the member attribute, dashed arrows represent the actual dependency relations).

Special functions to find parents and children in coordinated constructions were provided, but they worked well only if there were no errors in the annotations of the constructions.

Parent of an attribute: An attribute can have only a limited set of parents: nouns, some pronouns and numerals. Many exceptions exist to this rule, though, as almost any word can function as a name (e. g. *"the new Wash and Go"*).

Periods and abbreviations: Periods after abbreviations had to be annotated in a different way than those at the end of a sentence. Errors often implied wrong sentence segmentation (see above).

Prepositions and conjunctions: Prepositions and conjunctions can be detected either by their morphological tag, or by their analytical function. If these two indicators do not agree, there is probably an error.

4.4 Tectogrammatical Checking Procedures

Complement: Complement depends on two words: on a verb and its child (subject or object). Since the annotation scheme of PDT does not allow for "double dependencies", there were special rules how to deal with complements: on the a-layer, the parent of a complement should be the actant of the verb, if it exists. The analytical function of the complement in such a case is `Atv`. If the actant is not present, the parent of the complement should be the verb and the analytical function `AtvV`. The rules were different for the t-layer: the parent of a complement was always the verb and the relation to the actant was captured as a reference. However, the realisation had to be in accord: If the complement had the analytical function `Atv`, the reference should lead to the t-node corresponding to its analytical parent (see Figure 2). Otherwise, an added node should be the referent (see Figure 3).

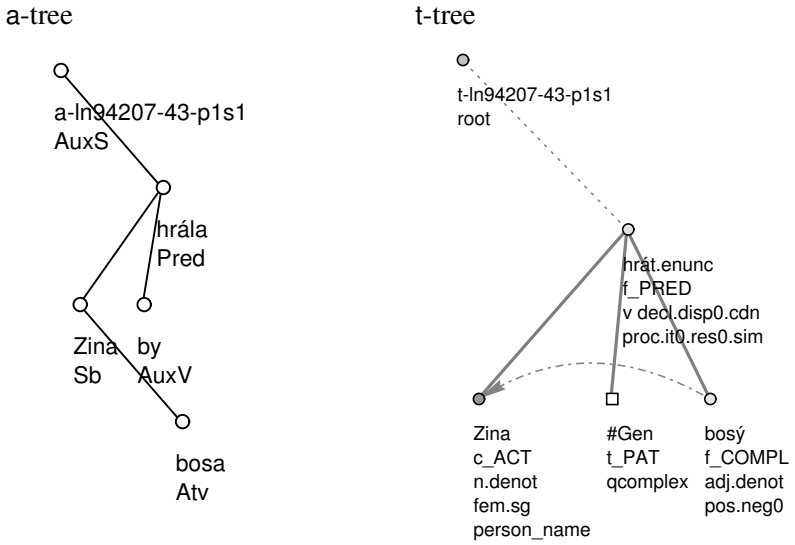

Figure 2: Complement with the analytical function `Atv`: *"Zina by hrála bosa." (Zina would play barefoot.)*

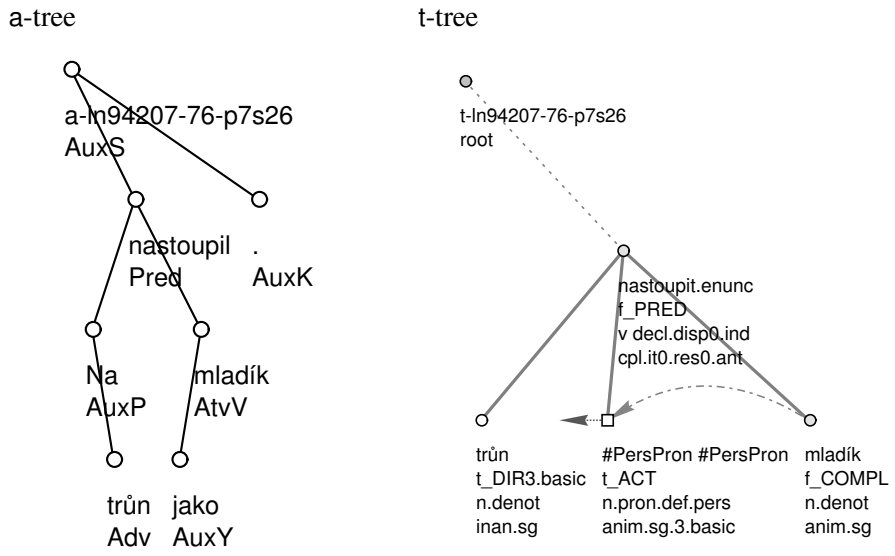

Figure 3: Complement with the analytical function `AtvV`: *"Na trůn nastoupil jako mladík." (He mounted the throne as a youngster [On throne he-mounted as youngster].)*

Lists of forms: For almost every functor there exists a limited set of possible forms.[4] This CP listed all forms of all functors so that the rarely occurring forms for each functor could be checked by a human annotator.

Comparison: Annotation rules for structures expressing comparison were quite complicated (for example, the sentence "Envy of bureaucrats is stronger than intellect" is annotated as if it would have been created with ellipsis from "Envy of bureaucrats is stronger than the intellect is strong", see Figure 4). Annotators often erred in such cases: they did not add all the missing words or corrupted the links from them to the a-layer.

New and cancelled functors: During the annotation process, some of the functors were cancelled, because they were found too specific and indistinguishable from some others. Similarly, some others were added, because an old functor was found too broad and general.

Coordination: Problems with coordination were similar to those at the a-layer. The situation was a bit more complex because it was possible for nodes with different functors to be coordinated. Therefore, it was less easy to guess the mistakes.

Parenthesis: All the parenthetical nodes were marked by a special attribute. When adding a new node under a parenthetical one the annotators often forgot to mark the new node as parenthetical, too.

Valency: Valency frames were captured as links to a valency lexicon. The CP had to verify that all the valency slots are filled with nodes of the prescribed form.

Reciprocity: In Czech, reciprocity can be expressed by several means (reflexive pronoun, collective words, or adverbs similar to English "each other").

[4] By form we mean morphological categories *and* auxiliary words (prepositions and conjunctions).

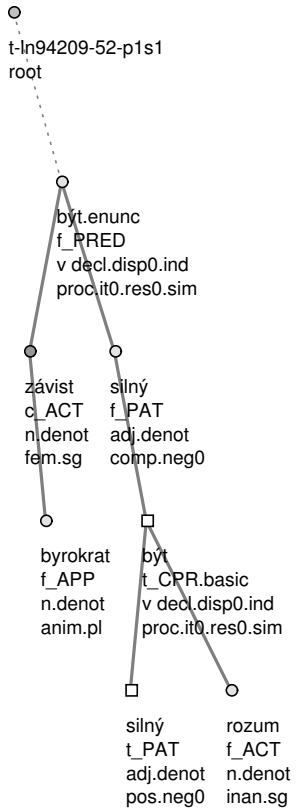

Figure 4: Comparison: *"Závist byrokratů je silnější než rozum" (Envy of-bureaucrats is stronger than intellect).*

Coreference: Coreference was represented by a link to the antecedent of a node. For some kinds of nodes, coreference was obligatory, which was consequently verified by the CP.

Relative and content clauses: Restrictive relative clauses starting with some pronouns can be distinguished from content clauses either by the functor of their head verb, or by the agreement between the pronoun and the parent of the sentence (restrictive) or by the agreement between the pronoun and its parent (content). These two aspects had to be in accord.

Topic-focus articulation: CP's for topic-focus articulation (TFA) (Sgall, Hajičová, and Panevová, 1986) made a special group of its own with almost thirty members. Due to the connection to deep word order, any change in the data that added new nodes or moved existing ones could damage the correctness of the TFA annotation.

CP's verified several invariants (every sentence must have a focus; topic is typically on the left side, while focus on the right and so on). Projectivity of trees was also checked by TFA related CP's because of its relation to deep word order.

Another area checked by TFA CP's were rhematisers – words that emphasise a part of a sentence and mark the focus or contrastive topic.

Annotators' comments: Annotators were allowed to write comments in a special attribute. Some of the comments had standard values and were amenable to be processed by a procedure, others had to be checked manually. All the comments had to be read and solved before the data were ready for release.

Grammatemes: Grammatemes correspond to morphological categories, but express their "semantic" values (e. g. *"trousers"* is plural morphologically, but its grammateme of number may be singular). Grammatemes were annotated manually only partially, most of them were generated by a program. The program was able to indicate some errors of various kinds if it was not able to assign any grammateme.

Names of people: All personal names were marked by a special attribute. Morphological information as well as capitalisation were used in heuristics with manual corrections.

Direct speech: All heads of a direct speech were marked by a special attribute. This attribute was used by CP's for valency annotation, because some verbs allow participants only in direct speech.

5 Quantitative Analysis of the Changes at the Tectogrammatical Layer

Some of the errors and changes in the data could have destructive character (e. g. deleting nodes). Therefore, older versions of the data were kept for later reconstruction. Only the tectogrammatical data were subject to this procedure because they changed most and most often.

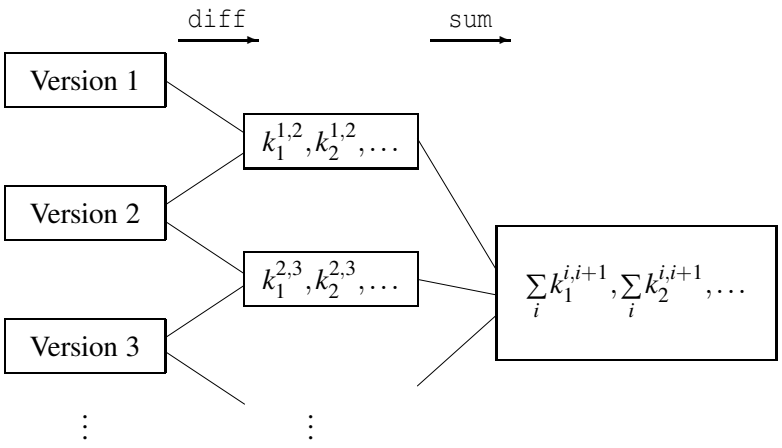

Figure 5: Diagram for obtaining the number of changes.

A special tool was created to compare particular versions of the data, showing the progress of the post-annotation checking process. The principle is shown in Figure 5: first, the number of changes between neighbouring versions of the data was counted (shown as the `diff` arrow) for various categories of the annotation. Finally, the numbers of changes were summed for all the categories (`sum` arrow, $k_i^{m,n}$ corresponds to the number of changes for the category i between versions m and n).

The resulting numbers had to be further corrected to give a realistic picture. Final numbers are presented in Table 1 (although the numbers are not rounded, they should not be taken as precise, because the influence of some factors was only estimated).

The = sign indicates that the value in the "all" column is the same as the value in the "relevant" column – i. e. all the nodes were relevant for the category. Numbers in parentheses are somehow "imprecise", because the total number of nodes was changing all the time and it is not clear how to count the percentage.

The @ sign means that the annotation of the category was merged to the data very lately, so the number of differences is close to the number of changes.

Category	Num. of changes	% all	% relevant	Num. of differences
New nodes	20 222	=	(2,4)	17 867
Deleted nodes	11 486	=	(1,4)	9 131
Form	2 281	=	0,3	723
Reason of form change	8 625	=	1,0	6259
m-lemma	12 957	=	1,5	8 886
Tag	29 573	=	4,0	22 796
a-structure	4 973	=	0,6	3 001
Analytical function	4 453	=	0,5	2 873
t-structure	37 865	4,1	5,6	32 642
Functor	37 330	5,2	5,5	31 918
t-lemma	25 609	=	3,8	22 085
Hidden	2 884	=	0,3	1 623
Unhidden	6 278	=	0,7	4 048
Coordination-like constructions	11 330	1,2	42,5	5 626
Links to a-layer	54 173	6,0	42,1	42 365
Coreference – links	6 348	0,7	13,4	@
Coreference – type	5 440	0,6	11,5	@
Topic-focus articulation	27 249	3,0	6,3	@
Deep word order	51 383	=	5,6	@

Table 1: Number of changes.

The last column corresponds to $k_i^{1,25}$, since there were 25 versions of the data.[5] The number of differences is always smaller than the number of changes: for some nodes, a category could change several times. The subtraction does not represent redundant work, though — it rather indicates that some parts of the data were really problematic and changed several times.

It is interesting (but not surprising) that numbers inside "super-categories", indicated by horizontal lines, are similar, while numbers from different super-categories vary. The only exception is m-tag: the reason lies in its information value, because the tag contains information about several morphological categories (gender, number, tense etc.).

6 Conclusion

The total number of changes is 361,136 (some lines in Table 1 are missing). If each node had been touched only once, the post-annotation process would have changed 43 % of the nodes. This number shows that the checking procedures did have their reason – not only they corrected annotation errors, but they also helped to keep the data in accordance with changing rules and to pick the preferred annotation in ambiguous cases.

In case of such a large and complex project as syntactically annotated corpus, choosing the abstract annotation scheme and data model is very important. Annotated phenomena should be captured not only in a way that describes all the aspects in a straightforward way and minimises the data volume, but also in a way that is easily understood by the annotators. The annotation tools should be able to validate the data to some extent to prevent the simple mistakes.

Iterative running of a set of checking procedures seems to be a good tool not only to guarantee the data validity and integrity, but also a good tool to measure the import of the procedures and the evolve-

[5]For some categories, the first version of the data could not be used, because the annotation was merged to the data in a later version.

ment of the data. Applying such procedures during the annotation process allows for the annotators' feedback and reduces the possibility of constant errors.

Finding a general way to generate checking procedures is almost impossible. Corpora differ in language (or languages) they capture, data formats they use, set of phenomena they describe, and theoretical framework they are built upon. However, comparing differing annotations of identical source data segments could give some hints on where to search for errors.

Attention should be paid to building the annotation guidelines and tools, so that the chance of a mistake is decreased – mainly in the areas of higher-level relations that are not easily presented to annotators. Relations between layers of annotation are typical example; moreover, they can be advantageously used to find annotation errors.

Acknowledgements

This paper was written with the support of the grant GA AV ČR 1ET101120503 (Integration of language resources in order to extract information from natural language texts).

References

Hajič, Jan, Eva Hajičová, Petr Pajas, Jarmila Panevová, Petr Sgall, and Barbora Vidová Hladká. 2001. Prague Dependency Treebank 1.0 (Final Production Label). CD-ROM. CAT: LDC2001T10.

Hajič, Jan. 1998. Building a Syntactially Annotated Corpus: The Prague Dependency Treebank. In Eva Hajičová, editor, *Issues of Valency and Meaning: Studies in Honour of Jarmila Panevová*. Karolinum - Charles University Press, Prague, pages 106–132.

Hajič, Jan, Marie Mikulová, Alla Bémová, Eva Hajičová, Jiří Havelka, Veronika Kolářová-Řezníčková, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Magda Razímová, Petr Sgall, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, and Zdeněk Žabokrtský. In preparation. The Prague Dependency Treebank 2.0. CD-ROM. http://ufal.mff.cuni.cz/pdt2.0/.

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

Květoň, Pavel. 2006. *Rule based morphological disambiguation*. Ph.D. thesis, MFF UK.

Ondruška, Roman, Jarmila Panevová, and Jan Štěpánek. 2003. An Exploitation of the Prague Dependency Treebank: A Valency Case. In Kiril Simov and Petya Osenova, editors, *Proceedings of the Workshop on Shallow Processing of Large Corpora (SproLaC 2003)*, pages 69–77, Lancaster, UK. UCREL, Lancaster University.

Pajas, Petr and Jan Štěpánek. 2005. A Generic XML-Based Format for Structured Linguistic Annotation and Its Application to Prague DependencyTreebank 2.0. Technical Report TR-2005-29, ÚFAL MFF UK, Prague, Czech Rep., December.

Sgall, Petr. 1967. *Generativní popis jazyka a česká deklinace*. Academia, Prague.

Sgall, Petr, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. Academia, Prague, Czech Republic.

Zeman, Dan, Jiří Hana, Hana Hanová, Jan Hajič, Barbora Hladká, and Emil Jeřábek. 2005. A Manual for Morphological Annotation, 2nd edition. Technical Report 27, ÚFAL MFF UK, Praha.