Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

Anotace víceslovných výrazů v Pražském závislostním korpusu

Pavel Straňák

Praha, 2010

Ústav formální a aplikované lingvistiky
I-3 Matematická lingvistika

Disertační práce byla vypracována v rámci doktorského studia na Ústavu formální a aplikované lingvistiky Matematicko-fyzikální fakulty Univerzity Karlovy v Praze v letech 2002 až 2010.

Uchazeč:	Pavel Straňák
Školitel:	Prof. RNDr. Jan Hajič, Dr.
Školicí pracoviště:	Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University in Prague
Oponenti:	Doc. PhDr. Karel Pala, CSc., Masarykova Univerzita, Fakulta informatiky, Botanická 68a 602 00 Brno Pavel Pecina, PhD. Centre for Next Generation Localisation School of Computing Dublin City University Glasnevin Dublin 9 Ireland
Předsedkyně OR I-3:	Prof. PhDr. Jarmila Panevová, DrSc.

Autoreferát byl rozeslán dne

Obhajoba se koná dne 23. září. 2010 v hodin před komisí pro obhajoby disertačních prací oboru I-3 Matematická lingvistika na Matematicko-fyzikální fakultě UK, Malostranské nám. 25, Praha 1, místnost S1 (4. patro).

S doktorskou disertační prací je možno seznámit se na studijním oddělení doktorského studia, Ke Karlovu 3, Praha 2.

Charles University in Prague
Faculty of Mathematics and Physics

Annotation of Multiword Expressions in the Prague Dependency Treebank

Pavel Straňák

Prague, 2010

Institute of Formal and Applied Linguistics
I-3 Mathematical Linguistics

The results comprised in the thesis were obtained within the candidate's doctoral studies at the Faculty of Mathematics and Physics, Charles University in Prague (MFF UK) during the years 2002–2010.

Candidate: Pavel Straňák

Supervisor: Prof. RNDr. Jan Hajič, Dr.

Department: Institute of Formal and Applied Linguistics,
Faculty of Mathematics and Physics,
Charles University in Prague

Opponents: Doc. PhDr. Karel Pala, CSc.,
Masaryk University,
Faculty of Informatics,
Botanická 68a
602 00 Brno
Czech Republic

Pavel Pecina, PhD.
Centre for Next Generation Localisation
School of Computing
Dublin City University
Glasnevin
Dublin 9
Ireland

I-3 Board Chair: Prof. PhDr. Jarmila Panevová, DrSc.

The summary was disseminated on

The thesis defence will be held on September 23, 2010 at in the building of MFF UK, Malostranské nám. 25, Praha 1, room S1 (4th floor).

The thesis is available at the Study and Students' Affairs Division, Ke Karlovu 3, Praha 2.

1 Introduction

In our project we annotate all occurrences of MWEs (including named entities, see below) in PDT 2.0. When we speak of **multiword expressions** we simply mean "idiosyncratic interpretations that cross word boundaries" (Sag et al., 2002). We do not inspect various types of MWEs, because in this project, we are not concerned with their grammatical attributes. We only want to identify them. Once there is a lexicon with them and their occurrences annotated in a corpus, the description and classification of MWEs can take place, but that is a new, different project.

We distinguish a special type of MWEs, for which we are mainly interested in its type, during the annotation: **named entities (NE)**. Treatment of NEs together with other MWEs is important, because syntactic functions are more or less arbitrary inside a NE (consider an address with phone numbers, etc.) and so is the assignment of semantic roles. That is why we need each NE to be combined into a single node, just like we do it with MWEs in general.

For the purpose of annotation we have built a repository of MWEs, which we call SemLex. We have built it using entries from some existing dictionaries, but it was significantly enriched during the annotation in order to contain every MWE that was annotated.

2 S-Data – The design and the PML schema

We decided to create a stand-off layer for any additional annotations that use nodes of existing trees and create some new units, while linking these new units to entries from an annotation lexicon. Since PDT 2 uses the PML format, our obvious choice was to design an additional PML layer.

s-data means s-layer PML files and the PML schema of these files. The idea behind s-data design is to have a simple way to store additional "sense" annotations over any layer of PDT. The annotations are stored as a set of "sense" nodes. Each s-node contains a link to a sense repository (annotation lexicon) and a set of references to nodes (m-, a- or t-) that correspond to an instance of the sense.

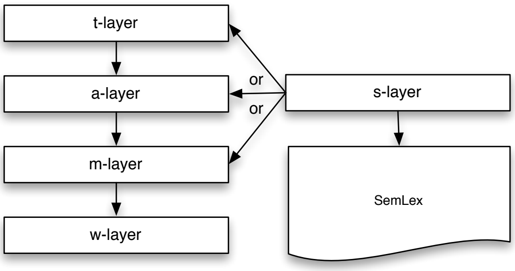


Figure 1: Relation of s-layer to PDT and SemLex

The design of s-data is quite universal. S-files can be used to provide additional annotations over any PML files that contain nodes thati have an attribute ID. The sense repository (annotation lexicon) can be any dictionary that provides IDs for the entries.

The PML schema of s-data is also not too complex: the elements `reference` in the beginning say that s-files can use references to nodes defined in m-data, a-data, or t-data, and in `SemLex`. Then there is a definition of the main structure of an s-file: the root element `sdata` with the child `meta` for metadata about the annotation and the child `wsd` for the annotation itself. The annotation, i.e. content of the `wsd` element, is defined as a sequence of `sm-, sa-`, or `st-nodes`. Those nodes are units that refer to nodes in m-, a- or t-files to define their extent, as described below. The whole sequence must contain nodes of only one of these types, because we cannot think of annotation that would require mixing references to m-nodes, a-nodes and t-nodes.

Listing 1: An st-node that identifies two nodes in a t-tree as a `SemLex` entry with ID `#institution` – a named entity of the type 'institution'.

```
<st id=”s−mf930709−001−161”>
  <lexicon−id>s##institution</lexicon−id>
  <tnode.rfs>
    <LM>t#t−mf930709−001−p3s1Bw14</LM>
    <LM>t#t−mf930709−001−p3s1Bw15</LM>
  </tnode.rfs>
</st>
```

3 Visualisation using SemAnn

The visualisation of annotated files in `SemAnn` (see Figure 3) has the advantage of showing whole text with all the MWEs clearly marked in a single glance. Seeing the whole text is very important, because context is crucial to distinguish some MWEs from isomorphic syntactic constructions that are fully transparent and have usually very different meaning. Seeing the MWE itself isolated, it may be quite challenging to come up with the meaning, even if one knows it immediately when the MWE is in context. Take *nohy postele* for example.[1]

Integration of the SemLex browser is also beneficial, because it allows fast and convenient lookup of annotated MWEs in `SemAnn`.

There are, however, also some drawbacks of this "full plain text of an article" approach:

[1] As a transparent syntactic construction, it means the legs of a bed. As an idiom it means the part of a bed, where one puts one's legs.

- It provides no way to directly compare two or more annotations.
- It is not efficient in case one needs to examine not only the annotation, but also the tectogrammatical tree structures, or any attributes of t-nodes.

4 Visualisation using TrEd

Figure 2 shows a tectogrammatical tree from a file that was annotated by two annotators. One of them identified two MWEs in this tree, the other only one. We can see that by looking at the patterns of the node groups (the "bubbles" around the groups of nodes). The crosscheck pattern is actually an overlap of two co-extensive node groups.

The colours used for node groups correspond to those used in SemAnn, but they can be easily redifined:[2]

```
my %mwe_colours = (
    semlex      => 'maroon',
    person      => 'olive drab',
    institution => 'hot pink',
    location    => 'Turquoise1',
    object      => 'plum',
    address     => 'light slate blue',
    time        => 'lime green',
    biblio      => '#8aa3ff',
    foreign     => '#8a535c',
    other       => 'orange1',
);
```

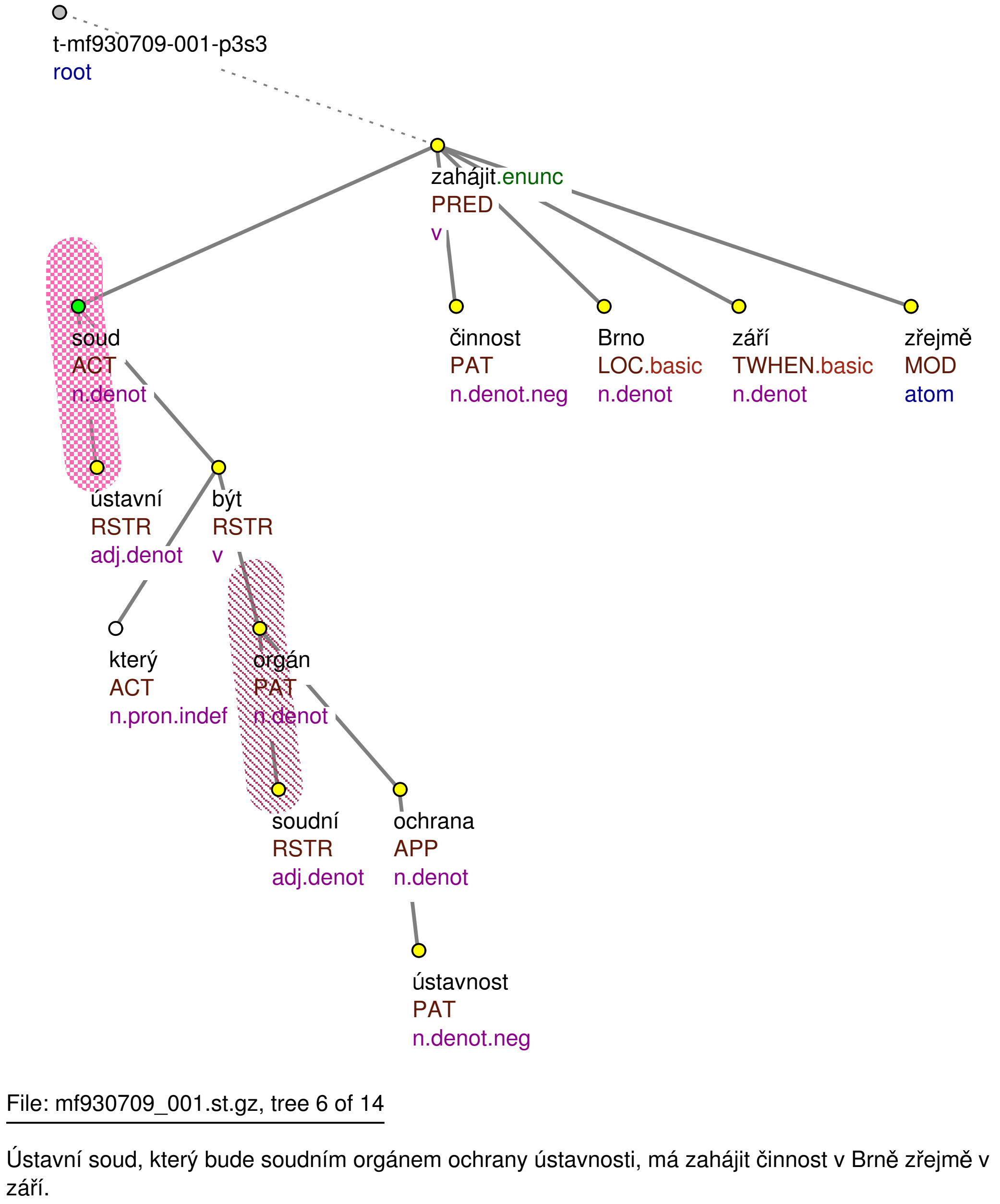
More information on technical aspects of this visualisation follows in the next section.

5 TrEd extension

TrEd has a powerful mechanism that allows it to be extended for new tasks. We developed an extension pdt-t-st that allows to see MWEs as graphically marked groups of tectogrammatical nodes. In order to do that we enhanced the t-data PML schema with information from s-files.

Main features of the extension:

[2] This is a quotation of the perl code from one of the source files of the TrEd extension: /pdt_t_st/contrib/pdt_t_st/display_mwe_groups.mak



Ústavní soud, který bude soudním orgánem ochrany ústavnosti, má zahájit činnost v Brně zřejmě v září.

Figure 2: MWEs displayed as node groups in a tectogrammatical tree. Different angles of a pattern distinguish annotators, thus the crosscheck pattern means the MWE was annotated by both. Colours distinguish `SemLex` entries (the expression *soudní orgán* and types of NEs (the expression *Ústavní soud* is of a NE type 'institution'.

- Merges the st-files into t-files and allows to display these enriched tectogrammatical trees.
- Types of annotated MWEs (i.e. types of NEs and `SemLex` entries) are distinguished with the same colours that were used in `SemAnn` during annotations. This allows not only for easily seeing NE types, but also easily spotting annotators' disagreement on them.
- Allows to merge annotations of several annotators into one t-file.
- Each annotator's MWEs have a unique raster. It is thus easy to spot annotators' partial or full disagreement not on types of MWEs, but also on their spans. See the MWE that was annotated by two annotators, and the one that was not in Figure 2.

There are two ways to merge the s-data and t-data:

1. Merge on opening the st-file in `TrEd`, and
2. Static merge that produces the merged `*.t.mwe.gz` file.

6 SemAnn

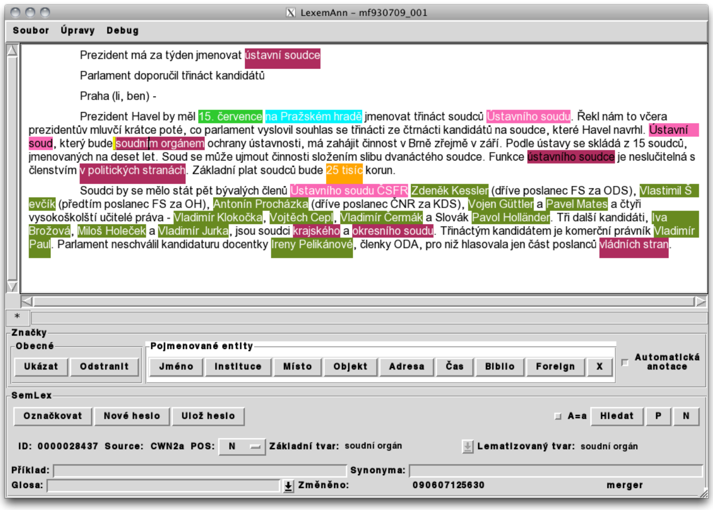
The annotation tool `SemAnn` is written in Perl 5[3] with Perl/Tk[4] GUI toolkit. The annotation tool depends on working installation of `TrEd`, specifically its unix installation, because it uses `nTrEd` for efficient execution of `TrEd` scripts in the background. `nTrEd` however, unlike `TrEd` itself or `bTrEd`, does not work on Windows.

`SemAnn` itself is composed of several main parts:

- The main application file `sem-ann.pl` mostly implements the application frontend. It implements the GUI, loads an s-file, a `SemLex`, and a log file for this s-file, if it had already been annotated. Then it takes care of all the interaction with the user and writes s-file, `SemLex`, and a log file.
- `n-TrEd` backend that is used to
  - generate surface sentences from tectogrammatical trees in t-files that are then displayed in the `SemAnn` GUI,
  - perform all the on-the fly pre-annotations (Section 8)
- The module `SemLex.pm` is used to read, save, query, and edit `SemLex`.

[3] `www.perl.org`; `dev.perl.org/perl5`

[4] `http://search.cpan.org/~srezic/Tk-804.029/`



Prezident má za týden jmenovat ústavní soudce

Parlament doporučil třináct kandidátů

Praha (li, ben) -

Prezident Havel by měl 15. července na Pražském hradě jmenovat třináct soudců Ústavního soudu. Řekl nám to včera prezidentův mluvčí krátce poté, co parlament vyslovil souhlas se třinácti ze čtrnácti kandidátů na soudce, které Havel navrhl. Ústavní soud, který bude soudním orgánem ochrany ústavnosti, má zahájit činnost v Brně zřejmě v září. Podle ústavy se skládá z 15 soudců, jmenovaných na deset let. Soud se může ujmout činnosti složením slibu dvanáctého soudce. Funkce ústavního soudce je neslučitelná s členstvím v politických stranách. Základní plat soudců bude 25 tisíc korun.

Soudci by se mělo stát pět bývalých členů Ústavního soudu ČSFR Zdeněk Kessler (dříve poslanec FS za ODS), Vlastimil Ševčík (předtím poslanec FS za OH), Antonín Procházka (dříve poslanec ČNR za KDS), Vojen Güttler a Pavel Mates a čtyři vysokoškolští učitelé práva - Vladimír Klokočka, Vojtěch Cepl, Vladimír Čermák a Slovák Pavol Holländer. Tři další kandidáti, Iva Brožová, Miloš Holeček a Vladimír Jurka, jsou soudci krajského a okresního soudu. Třináctým kandidátem je komerční právník Vladimír Paul. Parlament neschválil kandidaturu docentky Ireny Pelikánové, členky ODA, pro niž hlasovala jen část poslanců vládních stran.

ID: 0000028437 Source: CWN2a POS: N Základní tvar: soudní orgán Lematizovaný tvar: soudní orgán

Příklad: Synonyma:

Glosa: Změněno: 090607125630 merger

Figure 3: An annotated document in SemAnn. the yellow "selection tag" is barely visible on the word *soudním*, because over a different colour tag, selection has just a bezel. The SemLex entry that is displayed in the Semlex-part of the UI – *soudní orgán* – is the one used to annotate the selected word. The black font colour in two tags distinguishes automatically pre-annotated MWEs.

- The module SemLex_heslo.pm implements the SemLex entry: its structure, attributes and accessors.
- There is also a suite of miscellaneous scripts mostly for validation of annotated data, comparing and merging multiple annotations, merging annotators' SemLexes, computing reliability of annotations, and other small tasks related to annotation and managing the annotated data and SemLexes.

6.1 Annotation logs

An important, and as far as we know unique, feature of SemAnn is the design of annotation logs. As soon as an s-fileis loaded in SemAnn, a *.st.log file is created and every action taken henceforth, that modifies the s-file, is logged, together with a timestamp.

Logs are saved in YAML format and timestamps are human readable on purpose. Thus it is easy to visually inspect the logs in case of problems with sfs, e.g. data corruption. It was helpful on several occasions. However the main point of logs is different. We created them mostly to be able to gain some insight into the process of annotations.

7 Annotation

The annotation proceeded as follows: During the whole project we employed two annotators at a time. Originally they processed data in parallel. Later, when we had enough data to evaluate inter-annotator agreement, each annotator was assigned different data.

The data was divided into batches that were pre-annotated (see Section 8). Later batches had more pre-annotation than the earlier ones. This was done both in order to speed up the work while pre-annotation was being developed and to be able to evaluate impact of each pre-annotation type on the quality and speed of annotations.

We have completed annotation of the t-layer of the PDT with NEs and MWEs. All the MWEs that were annotated are part of annotation lexicon `SemLex`. A large part of data was annotated in parallel. Table 1 shows how much data was annotated by 1, 2, or 3 annotators in parallel, compared to the size of PDT (t-data). The last column just indicates that we have indeed annotated all the data of PDT 2.0 t-layer.

parallel annot.	1	2	3	PDT	2+3/PDT	*/PDT
t-files	1,288	1,412	465	3,165	59%	100%
t-nodes	248,448	343,834	82,683	674,965	63%	100%

Table 1: Annotated data

A total of 8,816 `SemLex` entries were used during annotations, 5,352 of those entries were created by annotators. All of these entries now have tree structures as part of their entries.

8 Pre-annotation

We employed four types of pre-annotation, only some of which are based on the assumption that all instances of a MWE share the same tree structure:

A) External pre-annotation provided by Milena Hnátková (see Hnátková, 2002). With each MWE a set of rules is associated that limits possible forms and surface word order of parts of a MWE. This approach was devised for corpora that are not syntactically annotated and is very time consuming.

B) Our one-time pre-annotation with those MWEs from SemLex that have been previously used in annotation, and as a result of that, they already have a tree structure as a part of their entry.

C) Dynamic pre-annotation as in B, only with the SemLex entries that have been recently added by an annotator.

D) When an annotator tags an occurrence of a MWE in the text, other occurrences of this MWE in the article are identified automatically.

   This is exactly what happens:

   1) Tree structure of the selected MWE is identified via `n-TrEd`
   2) The tree structure is added to the MWE's entry in SemLex
   3) All the sentences in the given file are searched for the same MWE using its tree structure (via `n-TrEd`)
   4) Other occurrences returned by `n-TrEd` are tagged with this MWE's ID, but these occurrences receive an additional attribute "auto", which identifies them (both in the s-files and visually in the annotation tool) as annotated automatically.

Pre-annotation (A) was executed once for all of the PDT. (B) is performed each time we merge MWEs added by annotators into the main SemLex. We carry out this annotation in one batch for all PDT files remaining to annotate. (C) is done for each file while it is being opened in the annotation environment. (D) happens each time the annotator adds a new MWE into SemLex and uses it to annotate an occurrence in the text. In subsequent files instances of this MWE are already annotated in step (C), and later even in (B).

After the pilot annotation without pre-annotation (D) we have compared instances of the same tags and found that 10.5% of repeated MWEs happened to have two different tree representations. In the final data it is 771 entries out of 8,816 entries that were used, i.e. 8.75%.

9 Measuring the inter-annotator agreement

During the annotations we employed four annotators. Below we give examples and describe parallel data of just one pair of annotators: (sid, vim).

The ratio of general named entities versus SemLex entries was approx. 52:48 for annotator sid and 50:50 in the case of annotator vim. This, and some other comparisons are given in Table 2. Both annotators processed 1090 files in parallel. The data consists of 350,177 tokens representing 284,029 t-nodes.

10 The measure – weighted kappa

In this section our primary goal is to assess whether with our current methodology we produce reliable annotations of MWEs. To that end we measure the amount of inter-annotator agreement that is above chance. Our attempt exploits *weighted kappa measure* κ_w Cohen (1968).

The reason for using a weighted measure is essential for our task: we do not know which parts of sentences are MWEs and which are not. Therefore annotators work with all words and even if they do not agree on the type of a particular MWE, it is still an agreement on the fact that this t-node is a part of some MWE and thus should be tagged. This means we have to allow for partial agreement on a tag.

type of MWE	sid	vim
SemLex entries – instances	9,427	9,477
- total entries used	4,472	4,067
Named Entities	10,208	9,621
- address	20	2
- biblio	4	14
- foreign	83	50
- institution	2,344	1,928
- location	619	700
- object	1,046	1,299
- other	1,188	1,498
- person/animal	3,246	3,232
- time	1,658	898

Table 2: Annotated instances of significant types of MWEs by annotators sid and vim

To account for partial agreement we divide the t-nodes into 5 classes c and assign each class a weight w_c as follows:

$c = 1$ If the annotators agree on the exact tag from SemLex, we get maximum information: $w_1 = 1$.

$c = 2$ If they agree that the t-node is a part of a NE or they agree that it is a part of some entry from SemLex, but they do not agree which NE or which entry, we estimate we get about a half of the information compared to when $c = 1$: $w_2 = 0.5$.

$c = 3$ If they agree that the t-node is a part of a MWE, but disagree whether a NE or an entry from SemLex, it is again half the information compared to when $c = 2$, so $w_3 = 0.25$.

$c = 4$ If they agree that the t-node is not a part of a MWE, $w_4 = 0.051$. This low value of w accounts for frequency of t-nodes that are not a part of a MWE, as estimated from data: Agreement on not annotating provides the same amount of information as agreement on annotating, but we have to take into account higher frequency of t-nodes that are not annotated:

$$w_4 = w_3 \cdot \frac{\sum annotated}{\sum not\ annotated} \approx 0.051.$$

We can see that two ideal annotators who agree on all their assignments could not reach high agreement measure, since they naturally leave some t-nodes without an annotation and even if they are the same t-nodes for both of them, this agreement is weighted by w_4. Now we can define the agreement which two ideal annotators reach as $\widehat{U}$ in Equation 1. If N is the number of all t-nodes in our parallel data and $n_{A \cup B}$ is the number of t-nodes annotated by at least one annotator, then we estimate $\widehat{U}$ as follows:

$$\widehat{U} = \frac{n_{A \cup B}}{N} + 0.051 \cdot \frac{N - n_{A \cup B}}{N} = 0.213. \qquad (1)$$

$c = 5$ If the annotators do not agree whether to annotate a t-node or not, $w_5 = 0$.

The numbers of t-nodes n_c and weights w per class c are given in Table 3.

	Agreement				Disagreement
	Annotated			Not annot.	
	Agr. on NE / SL entry				
	Full agr.	Disagr.			
class c	1	2	3	4	5
# of t-nodes n	31,290	2,864	1,555	235,739	11,790
weight w	1	0.5	0.25	0.051	0
$w_c n_c$	31,290	1,432	388.75	12,022	0

Table 3: The agreement per class and the associated weights for annotators sid a vim over the data they annotated in parallel (batches 04–17).

Now that we have estimated the upper bound of agreement $\widehat{U}$ and the weights w for all t-nodes we can calculate our version of weighted κ_w:

$$\kappa_w^U = \frac{A_o - A_e}{\widehat{U} - A_e} = \frac{D_e - D_o}{\widehat{U} - 1 + D_e}\ . \qquad (2)$$

A_o is the observed agreement of annotators and A_e is the agreement expected by chance (which is similar to a concept of baseline in measuring systems (parsers, taggers, etc.)). κ_w^U is thus a simple ratio of our observed agreement above chance and maximum agreement above chance. In equivalent (and often used) definition, D_o and D_e are observed and expected disagreements.

Weights w come into account in calculation of A_o and A_e.

We calculate A_o by multiplying the number of t-nodes in each category c by that category's weight w_c (see Table 3), summing these five weighted sums and dividing this sum of all the observed agreement in the data by the total number of t-nodes:

$$A_{o,sid,vim} = \frac{1}{N} \sum_{c=1}^{5} w_c n_c \doteq 0.162.$$

A_e is the probability of agreement expected by chance over all t-nodes. This means it is the sum of the weighted probabilities of all the combinations of all the tags that can be obtained by a pair of annotators. Every possible combination of tags (including not tagging a t-node) falls into one of the categories c and thus gets the appropriate weight w. (Let us say a combination of tags i and j has a probability p_{ij} and is weighted by w_{ij}.)

We estimated these probabilities from annotated data

$$A_{e,sid,vim} = \sum_{i}^{SemLex} \sum_{j}^{SemLex} \frac{n_{q_iA}}{N_A} \frac{n_{q_jB}}{N_B} w_{ij} \approx 0.047\ ,$$

where n_{q_iA} is the number of lexicon entry q_i in annotated data from annotator A and N_A is the amount of t-nodes given to annotator A. Here, the non-annotation is treated like any other label assigned to a t-node.

The resulting κ_w^U is then

$$\kappa_w^U = \frac{A_o - A_e}{\widehat{U} - A_e} \doteq 0.695.$$

11 Estimation of annotation intervals and speed

One of the reasons to implement detailed logs of all the annotations (see Section 6.1) was to allow detailed analysis of the time aspect of annotations.

In Figure 5 we can see some inter- and -intra annotator variance in speed. It seems there are some clear tendencies: annotators tend to have their own speed, as shown by the splines. They also show different amount of variance in speed, especially annotator *sid*'s speed is visibly more stable.

We wrote a script that tries to approximate "work intervals" as follows: It simply takes all logs in a given batch of annotated files, extracts the timestamps, transforms them into POSIX time (`http://en.wikipedia.org/wiki/Unix_time`), and sorts this list of integers. Then we try to approximate work intervals by setting two variables: $fluency$ and $start$. The default is $fluency = 300$, which means that as long as two timestamps are less than 300 seconds apart, it is considered a continuous

annotation type	3								
annotators	sid, vim								
batch number	04	05	06	07	08	10	11	12	13
number of files	89	72	85	87	45	3	69	50	69
κ_w^U	0.6714	0.7474	0.7289	0.7312	0.7029	0.6622	0.6162	0.6703	0.6804
annotation type	4								
annotators	sid, vim								
batch number	14	15	16	17					
number of files	99	124	146	152					
κ_w^U	0.6940	0.7196	0.7162	0.6703					
annotation type	8								
annotators	sta, vim								
batch number	21	34	35						
number of files	81	147	162						
κ_w^U	0.7576	0.6958	0.7361						

Table 4: Kappa per annotation type. These are all the data that were annotated in parallel.

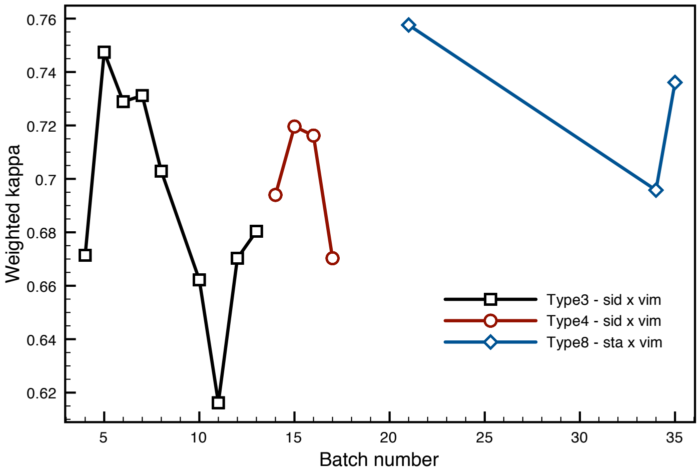


Figure 4: Weighted kappa per annotation type (colour line), a pair of annotators, and batches of annotated files (data points).

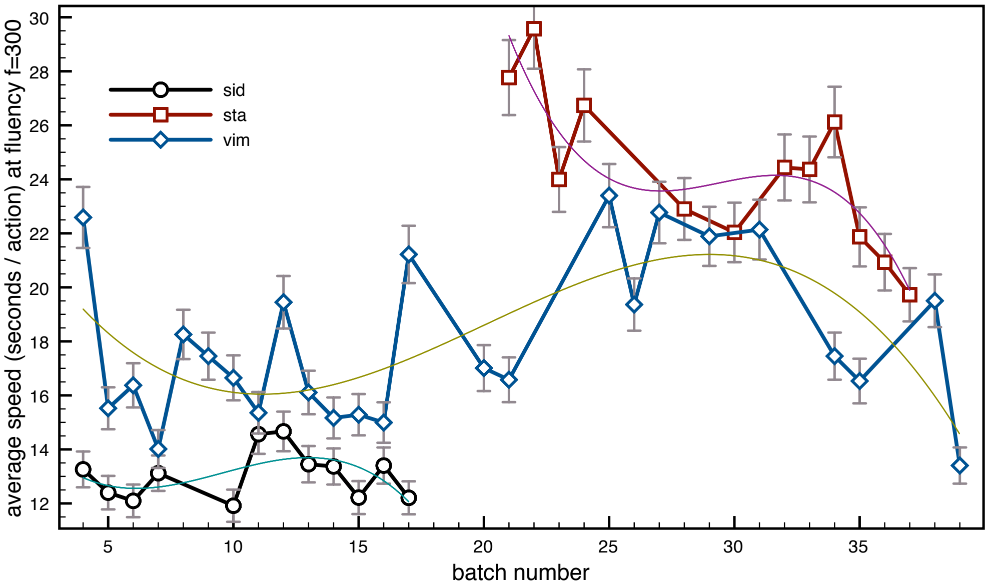


Figure 5: Average speed of each annotator for each batch that he/she annotated. Grey vertical bars show 5% error intervals.

(fluent) work. The value of $start$ is how much time we add to the length of interval of work on account of starting the work (start the computer, open the file, etc.) before first tag is logged. Default value is $start = 60$ (sec). So we split the list of timestamps into intervals, when there is at least 1 new timestamp every 5 minutes, add a minute to each interval and that gives us work intervals for each batch of files. We then divided the length of intervals by the number of timestamps in them to get an average speed of work in each interval. Finally average speeds counted over all intervals in a given batch of work are given in Figure 5.

12 Conclusions

In order to start with a move of t-lemmas PDT towards deeper representation of lexical meaning we needed to identify multiword expressions.

We came forward with two hypotheses based on the properties of dependency syntax and specifically of the tectogrammatical description: 1) That each MWE should form a single contiguous dependency structure, and 2) That all instances of a MWE should share the same dependency structure.

After examining a possibility of annotating t-trees directly we came with an idea of an annotation tool that presents a continuous plain text, but links the plain text to the underlying tectogrammatical structure, from which it is generated.

We proceeded to implement the annotation tool. As an integral part of the tool, we created a system of several types of pre-annotation of data. The most effective pre-annotation is based on the assumptions about tree structures of MWEs. We also devised a simple and efficient way for storing the annotation in a (relatively) human readable and still PML-compliant form by introducing *s-data*. As an important part of the annotation environment, we implemented detailed logs of the annotation that helped us to (at least to some extent) estimate the speed and price of annotation.

We also created a TrEd extension in order to be able to visualise and search s-data together with t-data in TrEd. The extension also provides means to create enriched t-layer that includes MWE annotation. This data can then be used for instance on a PML-TQ server without further dependency on the original s-data.

During our annotation two annotators at a time have annotated multiword expressions and named entities in the whole PDT 2.0 (t-layer). One of the annotators, who was with us for the whole duration of the project, actually annotated about half of the PDT herself.

One of the important result of the annotations is our annotation lexicon *SemLex*: It consists of all the MWEs identified during annotations. All SemLex entries contain tectogrammatical tree structures.

In Section 8 we show that the richer and the more consistent the tectogrammatical annotation, the better the possibilities for automatic pre-annotation that minimises human errors. In the analysis of inter-annotator agreement in Section 10 we show that a weighted measure that accounts for partial agreement as well as estimation of maximal agreement is needed. We present such a measure: κ_w^U. It is Cohen's weighted kappa with the upper bound moved from the value 1. This measure is the best fit for our problem that we were able to come up with.

The resulting κ_w^U has been gradually improving (cf. Bejček et al., 2008) as we were cleaning up the annotation lexicon, and employing more pre-annotation.

We have shown that our hypotheses about tree structures of MWEs hold, provided the tectogrammatical layer is correctly annotated. In this respect, our data, especially the places, where different t-structures were annotated with the same MWE from SemLex, also provide valuable information for both correcting errors and implementing new features in future versions of PDT.

The annotation tool `sem-ann` is freely available under a permissive licence. The annotated data and the annotation lexicon SemLex are also available and will be also published by the Linguistic data consortium. The TrEd extension is available to any TrEd user in the standard extensions repository and is available under the same permissive licence as `sem-ann`. For details on availability of tools, data, and licence, see `http://ufal.mff.cuni.cz/lexemann/mwe/`.

References

Eduard Bejček and Pavel Straňák. Annotation of multiword expressions in the prague dependency treebank. *Language Resources and Evaluation*, (44): 7–21, 2010. doi: 10.1007/s10579-009-9093-0. URL `http://www.aclweb.org/anthology-new/P/P09/P09-1002.pdf`.

Eduard Bejček, Petra Möllerová, and Pavel Straňák. The lexico-semantic annotation of PDT. In *Proceedings of the 9th International Conference, TSD 2006*, number 9 in Lecture Notes in Artificial Intelligence, pages 21–28. Springer-Verlag Berlin Heidelberg, 2006.

Eduard Bejček, Pavel Straňák, and Pavel Schlesinger. Annotation of multiword expressions in the prague dependency treebank. In *IJCNLP 2008 Proceedings of the Third International Joint Conference on Natural Language Processing* Asi (2008), pages 793–798.

Jacob Cohen. Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220, 1968.

Gregor Erbach and Brigitte Krenn. Idioms and Support Verb Constructions in HPSG. Technical report, Universität des Saarlandes, Saarbrücken, 1993.

Jan Hajič, Martin Holub, Marie Hučínová, Martin Pavlík, Pavel Pecina, Pavel Straňák, and Pavel Martin Šidák. Validating and improving the Czech WordNet via lexico-semantic annotation of the Prague Dependency Treebank. In *LREC 2004*, Lisbon, 2004.

Milena Hnátková. Značkování frazémů a idiomů v Českém národním korpusu s pomocí Slovníku české frazeologie a idiomatiky. *Slovo a slovesnost*, 2002.

Martin Holub and Pavel Straňák. Approaches to building semantic lexicons. In Jana Šafránková, editor, *WDC 2003*, pages 206–212, Praha, 2003. MATFYZPRESS.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for NLP. In *Computational Linguistics and Intelligent Text Processing: Third International Conference, CICLing*, volume 2276/2002 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, February 17-23 2002. URL `http://www.springerlink.com/content/k7etlqv25lxj3j1w/`.

List of Publications

Eduard Bejček and Pavel Straňák. Annotation of multiword expressions in the prague dependency treebank. *Language Resources and Evaluation*, (44): 7–21, 2010. doi: 10.1007/s10579-009-9093-0. URL `http://www.aclweb.org/anthology-new/P/P09/P09-1002.pdf`.

Pavel Straňák and Jan Štěpánek. Representing Layered and Structured Data in the CoNLL-ST Format. In Alex Fang, Nancy Ide, and Jonathan Weber, editors, *Proceedings of ICGL 2010*, pages 143–152. City University of Hong Kong, 2010. ISBN 978-962-442-323-5.

Ondřej Bojar, Pavel Straňák, and Daniel Zeman. Data issues in English-to-Hindi machine translation. In *Proceedings of LREC 2010*, 2010.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL-2009), June 4-5*, Boulder, Colorado, USA, 2009.

Eduard Bejček, Pavel Straňák, and Jan Hajič. Finalising multiword annotations in PDT. In *Proceedings of the 8th Conference on Treebanks and Linguistic Theories*, pages 17–25. Università Cattolica del Sacro Cuore, 2009. ISBN 978-88-8311-712-1.

Ondřej Bojar, Pavel Straňák, Daniel Zeman, Gaurav Jain, Michal Hrušecký, Michal Richter, and Jan Hajič. Obtaining mediocre results with bad data and fancy models. In *Proceedings of ICON 2009: 7th International Conference on Natural Language Processing*, pages 316–321. NLP Association of India, 2009.

Ondřej Bojar, Pavel Straňák, and Daniel Zeman. English-hindi translation in 21 days. In Sriram Venkatapathy and Karthik Gali, editors, *Proceedings of the 6th International Conference on Natural Language Processing (ICON-2008) NLP Tools Contest*, Pune, India, 2008. NLP Association of India, International Institute of Information Technology, Hyderabad.

Eduard Bejček, Pavel Straňák, and Pavel Schlesinger. Annotation of multiword expressions in the prague dependency treebank. In *IJCNLP 2008 Proceedings of the Third International Joint Conference on Natural Language Pro-*

cessing, pages 793–798. Hyderabad, India, 2008. Asian Federation of Natural Language Processing.

Eduard Bejček, Petra Möllerová, and Pavel Straňák. The lexico-semantic annotation of PDT. In *Proceedings of the 9th International Conference, TSD 2006*, number 9, pages 21–28, 2006.

Jan Hajič, Martin Holub, Marie Hučínová, Martin Pavlík, Pavel Pecina, Pavel Straňák, and Pavel Martin Šidák. Validating and improving the Czech WordNet via lexico-semantic annotation of the Prague Dependency Treebank. In *LREC 2004*, Lisbon, 2004.

Martin Holub and Pavel Straňák. Approaches to building semantic lexicons. In Jana Šafránková, editor, *WDC 2003*, pages 206–212, Praha, 2003. MATFYZPRESS.

Pavel Straňák. O čem mluvíme? (what do we talk about?). Master's thesis, Ostravská univerzita, Ostrava, Česká Republika, April 2001.