Annotation of Multiword Expressions in the Prague Dependency Treebank

Pavel Straňák

Doctoral Thesis

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

Praha, 2010

Author: 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,
Botanicka 68a
602 00 BRNO
Czech Republic

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

Contents

List of Figures

List of Tables

Abstract

This thesis explores annotation of multiword expressions in the Prague Dependency Treebank 2.0. We explain, what we understand as multiword expressions (MWEs), review the state of PDT 2.0 with respect to MWEs and present our annotation. We describe the data format developed for the annotation, the annotation tool, and other software developed to allow for visualisation and searching of the data. We also present the annotation lexicon SemLex and analysis of the annotation.

Chapter 1

Introduction

1.1 Motivation

Various projects involving lexico-semantic annotation have been ongoing for many years. Among those there are the projects of word sense annotation, usually for creating training data for word sense disambiguation. However majority of these projects have only annotated very limited number of word senses (cf. Kilgarriff 1998). Even among those that aim towards "all words" word-sense annotation, multiword expressions (MWE) are not annotated adequately (see Mihalcea 1998 or Hajič et al. 2004), because for their successful annotation a methodology allowing identification of new MWEs during annotation is required. Existing dictionaries that include MWEs concentrate only on the most frequent ones, but we argue that there are many more MWEs that can only be identified (and added to the dictionary) by annotation.

There are various projects for identification of named entities (for an overview see Ševčíková et al., 2007). We explain below, mainly in Section 1.2, why we consider named entities to be concerned with lexical meaning. At this place we just wish to recall that these projects only select some specific parts of text and provide information only for these. They do not aim for full lexico-semantic annotation of texts.

There is also another group of projects that have to tackle the problem of lexical meaning, namely treebanking projects that aim to develop a deeper layer of annotation in addition to a surface syntactic layer. This deeper layer is generally agreed to concern lexical meaning. To our best knowledge, the lexico-semantic annotations still deal with separate words, phrases are split and their parts are connected with some kind of dependency. Furthermore, only words with valency are involved in projects like NomBank (Meyers et al., 2004), PropBank (Palmer et al., 2005) or PDT-VALLEX (Hajič et al., 2003).

1.2 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), see Section 2. We do not inspect various types of MWEs, because we are not concerned in 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)**.[1] 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. We explain this in detail in Chapters 6 and 7.

[1] NEs can in general be also single-word, but in this phase of our project we are only interested in multiword expressions, so when we say NE in this paper, we always mean multiword.

Chapter 2

Multiword expressions

Baldwin (2004) defines MWEs very broadly as entities that are:

- "decomposable into multiple simplex words," and
- "lexically, syntactically, semantically, pragmatically and/or statistically idiosyncratic."

His examples are as follows: *"San Francisco, ad hoc, by and large, Where Eagles Dare, kick the bucket, part of speech, in step, the Oakland Raiders, trip the light fantastic, telephone box, call (someone) up, take a walk, do a number on (someone), take (unfair) advantage (of), pull strings, kindle excitement, fresh air, ..."*

From the definition and the examples it is clear that Baldwin includes not only idioms and complex verbs, but also any named entities and even any statistically or pragmatically important[1] collocations. At least that is what we understand as "statistically idiosyncratic". Such expressions include "environmental policy" but also "salt and pepper", which is semantically quite compositional and simple, but statistically the order of its components is significant. In the Corpus of Contemporary American English (COCA, `http://www.americancorpus.org/`), there are 3648 occurrences of "salt and pepper" vs. 62 occurrences of "pepper and salt". Of the 62 occurrences 60 are in recipes. This is rather extreme case of "statistical idiosyncrasy"; as such it well illustrates the point.

Such a broad definition basically says that MWEs are "interesting collocations" but in its broadness it is not suitable for our purpose. We are more interested in the more conventional approach that Baldwin has in most of his other (co-authored)

[1] We avoid a MWE (sic!) "statistically significant" on purpose, because we assume that Baldwin also avoids it on purpose when using a word "idiosyncratic". As far as we know "statistically idiosyncratic" is not a well defined term byt we understand it as saying that not any statistically significant difference in distribution is peculiar enough to be called "idiosyncratic". We are fully aware how imprecise this sounds.

papers (Baldwin et al., 2003; Sag et al., 2002). MWEs are viewed as "cohesive lexemes that cross word boundaries". This seems to be the most common definition of MWEs in NLP, as long as we abstract from differences in terminology.

There are of course many definitions of multiword expressions, multiword lexemes, phrasemes, idioms, and many other concepts that are more or less closely related. They are classified by many criteria, often into fine hierarchies of types and subtypes. Several of the important classifications are presented in Pecina (2009). For an exhaustive treatment of the issue see Čermák (2010).

We shall not go into detail of these classifications, however, because the rather intuitive view of MWEs as presented above is rather close to our view: idiosyncratic expressions that cross word boundaries. This way the definition is broad enough to account for idioms, as well as multiword terminology, and also many, though not all, named entities, as discussed in Section 2.1.

2.1 Named entities

Named entities are a concept originating in information extraction (Jurafsky and Martin, 2008). This concept is well rooted in NLP but it does not exist in classical lexicology and its defining criteria do not correspond with a definition of a lexical unit, lexeme, or any other lexicological or lexicographic concept of our knowledge.

We use the NE classification by Ševčíková et al. (2007) as a starting point and modify it a bit (see Section 6.1). However the definition of what named entities are is problematic:

> *„Pojmenované entity jsou jednotlivá slova nebo slovní spojení, která v textu vystupují jako pojmenování osob, míst, firem apod. Cílem anotace je označení všech pojmenovaných entit v předloženém lineárním textu. Současná anotace bude zaměřena na identifikaci především těch pojmenovaných entit, které jsou zapsány s velkým počátečním písmenem.“*

The definition basically says that named entities are single- or multi-word units that are used to name persons, locations, firms, etc. I.e. named entities are the expressions used to name entities. It is actually hard to call this a definition at all.

It is nevertheless not easy to define named entities properly. Other authors do not fare much better. For instance Jurafsky and Martin (2008) say that "By **named entity**, we simply mean anything that can be referred to with a proper name." It is practically the same definition as above. They further explain that for practical concerns the notion of NEs is often extended "to include things that aren't entities per se, but nevertheless have practical importance and do have characteristic signatures that signal their presence; examples include dates, times, named events, and other kinds of temporal expressions, as well as measurements, counts, prices, and other kinds of numerical expressions."

We believe we can now provide at least some additional constraints to the definition by Ševčíková et al. above: To be considered a named entity an expression must share at least some *features of idioms*: it cannot be an exploitation (see Hanks, 2010), i.e. it must fulfil some criteria of stability. However, unlike idioms, NEs can still vary significantly in form. An address can be tweaked in many small ways, still being the same address. So the stability of NEs can be described as normality: observance of common "design patterns", or "signatures", as they are called by Jurafsky and Martin.

Chapter 3

How are things in PDT 2.0

3.1 Prague Dependency Treebank

We work with the Prague Dependency Treebank (PDT, see Hajič, 2005), which is a large corpus with rich annotation on three layers: it has in addition to the morphological and the surface syntactic layers also the tectogrammatical layer. (In fact, there is also one non-annotation layer, representing the "raw-text" segmented into documents, paragraphs, and tokens.) Annotation of a sentence on the morphological layer consists of attaching several attributes to the tokens of the w-layer, the most important of which are morphological lemma and tag. A sentence at the analytical layer is represented as a rooted ordered tree with labeled nodes. The dependency relation between two nodes is captured by an edge with a functional label. The tectogrammatical layer has been construed as the layer of the (literal) meaning of the sentence and thus should be composed of monosemic lexemes and the relations between their occurrences.[1]

On the tectogrammatical layer only the autosemantic words form nodes in a tree (t-nodes). Synsemantic (function) words are represented by various attributes of t-nodes. Each t-node has a lemma: an attribute whose value is the node's basic lexical form. Currently t-nodes, and consequently their t-lemmas, are still visibly derived from the morphological division of text into tokens. This preliminary handling has always been considered unsatisfactory in FGD.[2] There is a clear goal to distinguish t-lemmas through their senses, but this process has not been completed so far.

[1] With a few exceptions, such as personal pronouns (that refer to other lexeme) or coordination heads.

[2] Functional Generative Description (FGD, Sgall et al., 1986; Hajičová et al., 1998) is a framework for systematic description of a language, that the PDT project is based upon. In FGD units of the t-layer are construed equivalently to monosemic lexemes and are combined into dependency trees, based on syntactic valency of the t-nodes.

Figure 3.1 shows the relations between the neighboring layers of PDT.

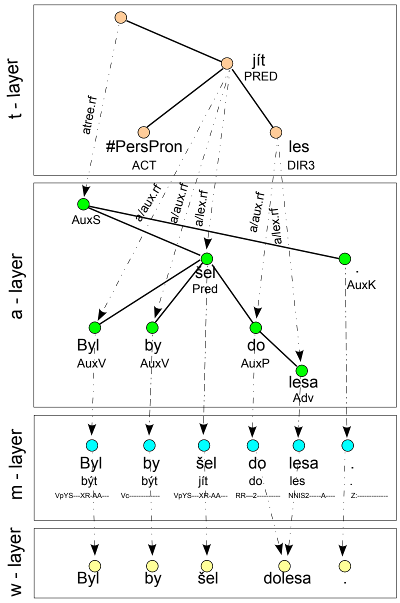


Figure 3.1: The rendered Czech sentence *Byl by šel dolesa.* (lit.: He-was would went toforest.) contains past conditional of the verb "jít" (to go) and a typo "toforest" repaired on m-layer.

Our project aims at improving the current state of t-lemmas. Our goal is to assign each t-node a t-lemma that would correspond to a lexeme, i.e. that would really distinguish the t-node's lexical meanings. To achieve this goal, in the first phase of the project, which we report on in this paper, we *identify multiword expressions and create a lexicon of the corresponding lexias*. A simple view of the result of our annotations is given in the Figure 3.2, some technical details are in Section 7.4.

In the Prague dependency treebank version 2.0 (Hajič et al., 2006) there are several functors that refer to multiword expressions (MWEs) in one way or another. There are also two technical lemmas `#Idph` and `#Forn` that identify roots of subtrees representing MWE's. Tectogrammatical annotation is described in detail in Mikulová et al. (2006).

3.2 Current state of MWEs in PDT 2.0

During the annotation of valency, which is a part of the tectogrammatical layer of PDT 2.0, the t-lemmas, have been basically identified for all the verbs and some nouns and adjectives. The resulting valency lexicon is called PDT-VALLEX (Hajič

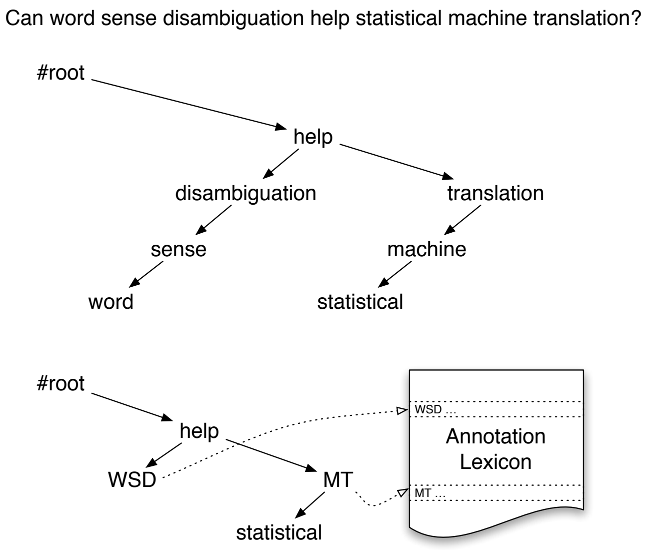


Figure 3.2: Schema of the changes in t-trees after integration of our annotations; every MWE forms a single node and has its lexicon entry

et al., 2003) and we can see it as a repository of lexemes based on verbs, adjectives and nouns in PDT that have valency. [3]

This is a starting point for having t-nodes corresponding to lexemes. However in the current state it is not fully sufficient even for verbs, mainly because parts of MWEs are not joined into one node. Parts of frames marked as idiomatic are still represented by separate t-nodes in a tectogrammatical tree (e.g. nodes with t-lemmas “`co`” in Figure 3.3 or “`k_dispozici`” in Figure 3.5). Verbonominal phrasemes are also split into two nodes, where the nominal part is governed by the verb. Non-verbal idioms have not been annotated at all in the current state of PDT.

In Figures 3.3, 3.4, and 3.5 we give several examples of t-trees in PDT 2.0, that include idioms, light verb constructions and named entities:

[3] It is so because in PDT-VALLEX valency is not the only criterion for distinguishing frames (=meanings). Two words with the same morphological lemma and valency frame are assigned two different frames if their meaning differs.

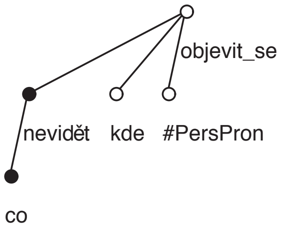


Figure 3.3: Idiom *Co nevidět* meaning “in a blink (of an eye)”, (literally: what not-see)

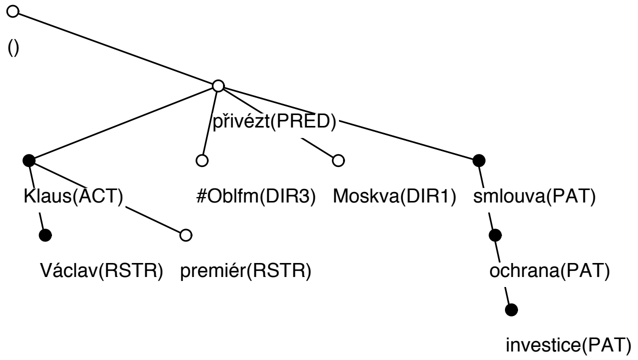


Figure 3.4: A sentence featuring a personal name and a name of a bilateral treaty (which is not the exact official name, however, thus it is not capitalised)

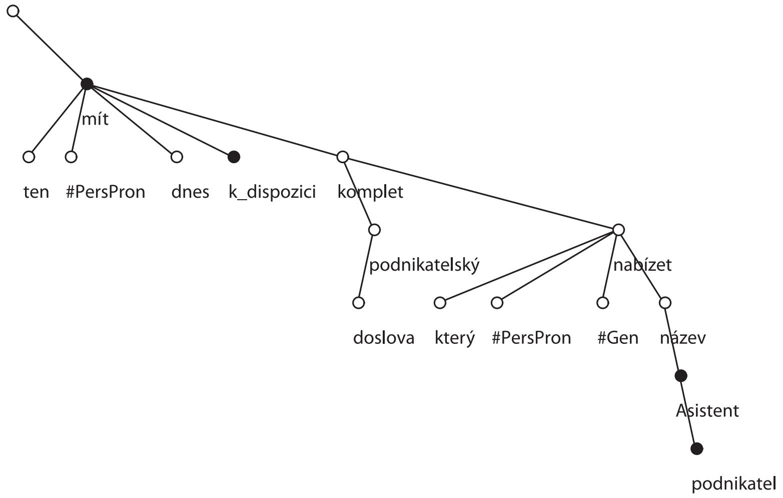


Figure 3.5: A t-tree of a sentence featuring a light verb construction *mít k dispozici* (lit.: to have at [one's] disposal) and a named entity (a product name*Asistent podnikatele* (lit.: assistant of-businessman) that looks like a common phrase, except for the capital 'A'.

Chapter 4

S-Data

4.1 Introduction

When we faced the prospect of creating annotations of MWEs in the PDT, we already knew that we wanted to work with t-layer, as described in Section 1.1. We were however reluctant to add our data directly into t-files. The principal reason was our uncertainty whether this information really belongs to the tectogrammatical layer of description. There were also secondary, but all the more practical reasons: the t-files are rather complex and we wanted a simple way to isolate our annotations. Also, we prefer to keep the stable PDT 2 as is and clearly separate our experiments from this stable data.

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

4.2 PML – Prague Markup Language

PML is a language designed by Pajas and Štěpánek (2005) for structured linguistic annotation. It can be used equally well for speech data (Hajič et al., 2008), text corpora annotated using dependency syntax, phrase-structure trees, or even both together as different layers of annotation over single underlying data (cf. Cinková et al., 2009). Dictionaries can also be represented in PML, e.g. PDT-Vallex – the valency dictionary that is a part of the PDT 2.0 (Hajič et al., 2003).

PML is a XML language, which means it can take advantage of the rich existing XML tools, above else parsers and validators. PML itself however defines in addition many data types and a system of roles. To allow for efficient design and validation of PML files, there is PML Schema. PML Schema files themselves can be validated using RelaxNG. The schema of PML workflow is illustrated in Figure 4.1. The full set of tools for working with PML data was published as the PML Toolkit (Pajas and Štěpánek, 2009).

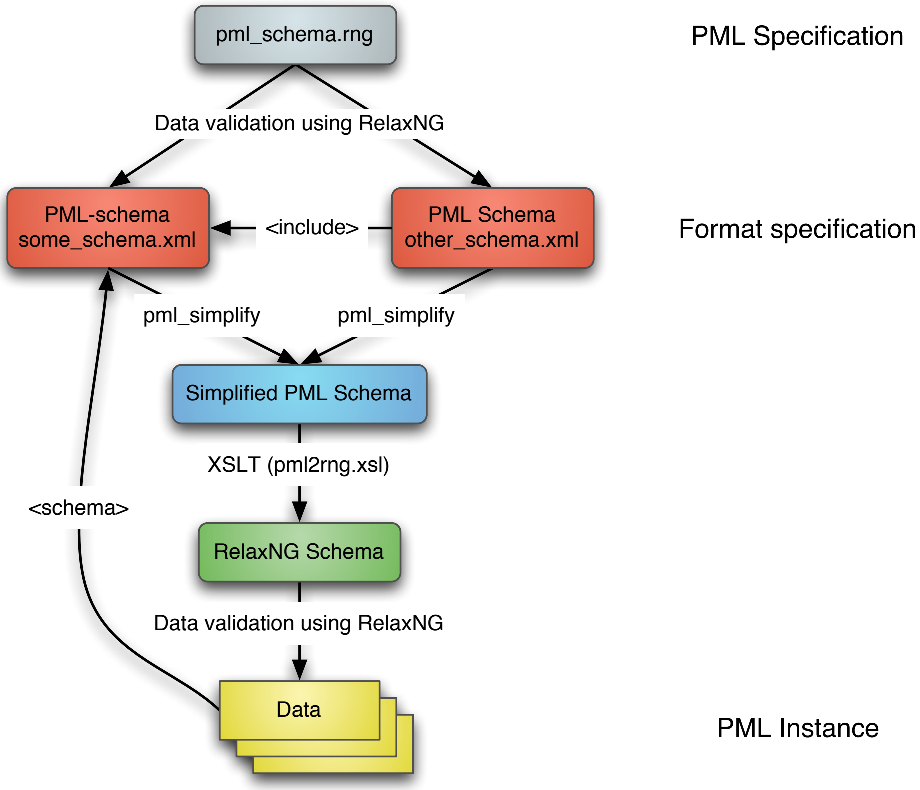


Figure 4.1: A schema of a PML workflow. It does not illustrate all the possible interactions of PML data and schema files.

4.3 The design and the PML schema of s-data

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.[1] An s-file is thus

[1] Although we have created the PML schema of s-layer primarily for annotations of MWEs, we made it quite generic. It can be utilized for any treebank annotations that use a large lexicon. For instance one s-file can contain multiple annotations of valency referencing to different valency

basically a simple flat list of s-nodes. It does not contain any trees. A single s-file can only reference a single PDT file: either tectogrammatical, or analytical, or even morphological layer can be used, but references to different layers cannot be mixed in one s-file. Figure 4.2 shows a relation of s-layer to PDT layers and SemLex.

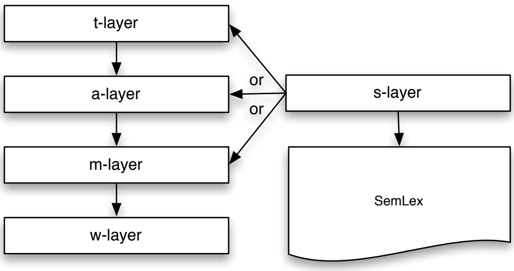


Figure 4.2: Relation of s-layer to PDT and SemLex

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 tools used in our annotations mostly expect PDT PML or the particular s-files that we have used, but that is mostly for convenience. Should the need appear to adapt the workflow a different corpus represented by PML files and a different annotation lexicon, the changes required would be rather minor.

The PML schema of s-data (see the code listing 4.4) 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.

After defining the structure of s-files, the schema defines the node types mentioned above. In order to do that, there is first a definition of a *generic s-node*. This generic s-node cannot be used in annotations directly, since it was not named in the definition of the element `wsd`, as described above. Thus we can see the definition of s-node as a description of common features of the *real* s-nodes: an s-node must have attributes `ID` and `src` and an element `lexicon-id`, that contains an ID or other unique identifier of an entry in an annotation lexicon, i.e. `SemLex` in our case (but it can be a different lexicon, in different data format, so the value of `lexicon-`

dictionaries. This generic nature of s-layer is the reason why it allows references to morphological, analytical or tectogrammatical layer of PDT, even though in our current project we only need the references to t-layer.

id can be for instance a reference to a node in a PML file of a lexicon).

The specific s-nodes: sm-, sa-, and st-node are defined next. The mechanism used for these definitions is called *derivation* and is similar for instance to type classes of some programming languages. It allows to define a generic type and then derive its more specific sub-types. All three definitions are almost identical, so we shall look only at the definition of the st-node:

The element derive defines the type st-node as a subtype of s-node. St-node thus inherits all that has already been defined for s-node and only extends the definition. The rest of the derivation defines that every node of the type st-node must have a main element <st-node> and it must also (in addition to 'ID', 'src', and 'lexicon-id') contain an element <tnode.rfs> that shall contain a list of PML references. An example of a single st-node is given in Figure 4.1. A short s-data file from our annotations is given in Listing 4.5 to provide full example including metadata and a list of annotated MWEs (i.e. st-nodes).

Listing 4.1: 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-l61”>
  <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>
```

4.4 Visualisation

There are two basic ways to view st-nodes: in SemAnn or in TrEd. Both of these need to use the "t-a-m-w-" PDT files to display the sentence and/or the tree for each sentence and then they read the st-file to add the information about st-nodes. The st-nodes are displayed as colour boxes or bubbles over the words in a sentence or nodes in a tree in SemAnn or TrEd respectively.

PML-TQ server may seem like an obvious third choice for the visualisation, but currently it is not the case. Since PML-TQ server uses TrEd for the visualisation of trees, the SVG graphic representation of a tree in PML-TQ client is actually generated by b-TrEd server running on the PML-TQ server. The problem is that TrEd does currently use bitmap patterns in addition to colours to distinguish between node groups. The patterns are then not exported into SVG and the result is that in our particular annotation we can see only partial information. While keeping the distinction of NE types and SemLex entries, we loose the information on annotators. There is also no easy way to tell whether the extent of the node group is correct, because in case annotators disagreed and one annotated nodes AB and the other BC, the node groups would merge into ABC. That is why currently, until for instance opacity is used to represent the information from patterns during SVG export in TrEd, PML-TQ server is not a suitable visualisation tool for our annotations.

4.4.1 Visualisation using SemAnn

The visualisation of annotated files in SemAnn 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.[2]

Integration of the SemLex browser is also beneficial, because it allows fast and convenient lookup of annotated MWEs in SemAnn. Details of SemAnn interface are described in Section 5.1.

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

- 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.

4.4.2 Visualisation using TrEd

Figure 4.3 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:[3]

[2] 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.

[3] 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

```
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.

4.5 `TrEd` extension

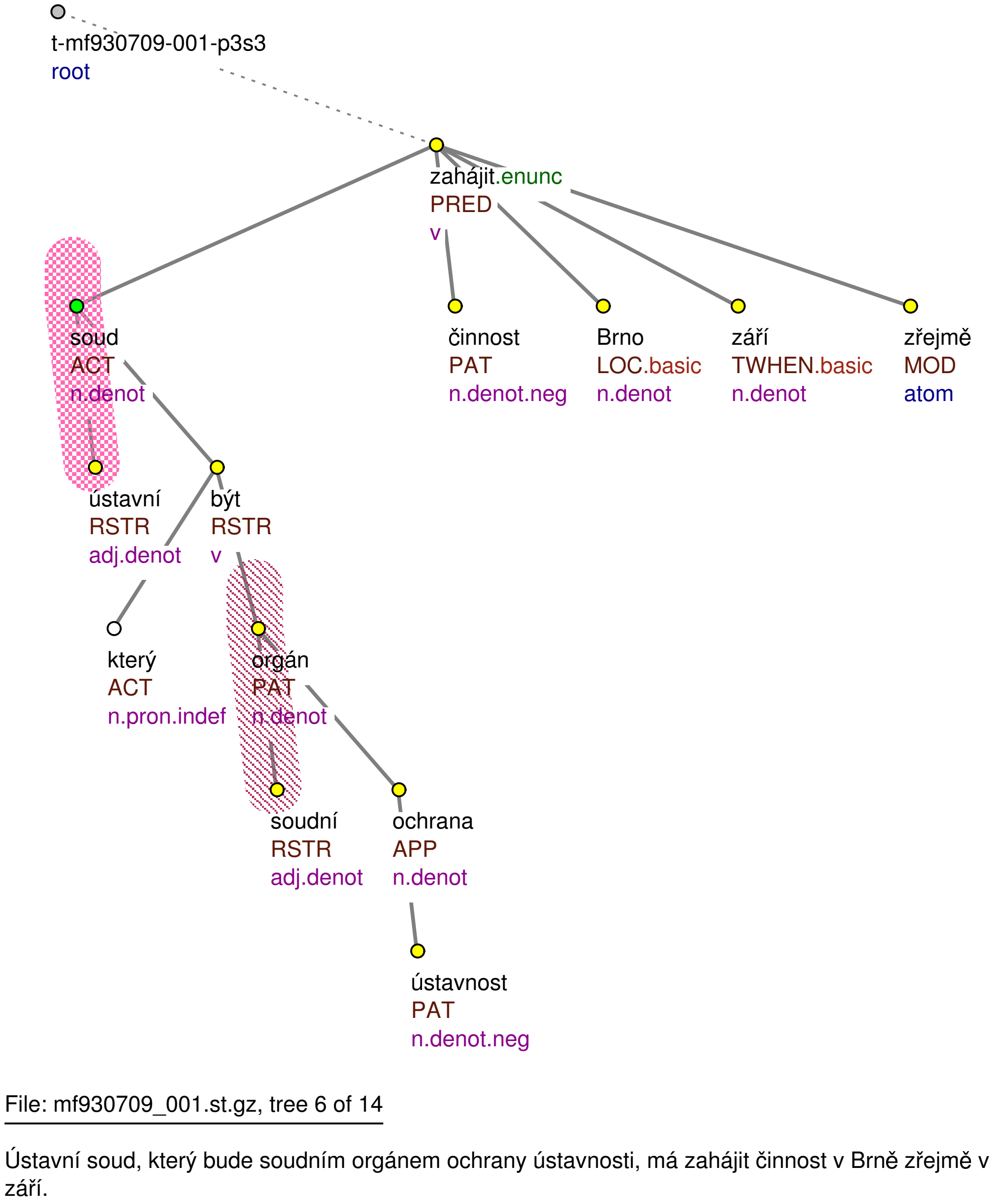
`TrEd` has a powerful mechanism that allows it to be extended for new tasks. We developed an extension `pdt-t-st` that allows to see MWEs as graphically marked groups of tectogrammatical nodes. In order to do that we enhanced the t-data PML schema with information from s-files. For details on the resulting t-mwe-data PML schema see the item `tdata_mwe_schema.xml` on page 29 and Listing 4.3.

Main features of the extension:

- 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 4.3.

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

1. Merge on opening the st-file in `TrEd`, and



File: mf930709_001.st.gz, tree 6 of 14

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

Figure 4.3: 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'.

2. Static merge that produces the merged `*.t.mwe.gz` file.

The dynamic merging is done using a newly developed feature of `TrEd`[4] that allows to apply arbitrary Perl transformations on the input data. To see, how this is done, see Listing 4.2. Thus we open the st-file, use the mechanism of extensions to activate our extension by identifying the st-file as data the extension can process and call our transformation. The transformation requires a t-file annotated by this s-file to be present in the same directory. The t-file and s-file are parsed, and for each st-node we find a tectogrammatical tree that includes t-nodes annotated (i.e. referenced) in this st-node. When we have a root of the correct t-tree, the st-nodes are basically added into an attribute `mwes` of this t-root. The attribute is rather complex, because it contains lists of st-nodes for all annotators that annotated any st-nodes in this tree. Some small transformations of st-nodes are needed, as well as creation of some new XML nodes, to represent the information from s-files in the t-files properly. For all the details inspect the code of `<tred-extensions-dir>/pdt_t_st/libs/SDataMerge.pm`.[5]

Full structure of the extension is displayed in Figure 4.5. We shall briefly describe the most important components, with emphasis on the bits of information that are not ideally documented.[6]

The extension can be used either in `TrEd`, as intended, for the most part, or its merging functionality can be invoked without `TrEd` by using the files in `<ext>/bin/`:[7]

`merge-s-and-t-layer.pl` – Integrates the s-layer annotation into the t-layer files. T-files must be in the same directory as the s-files. Runs the merge, that is actually implemented in the module SDataMerge.pm. This script is used to generate t-mwe-files statically. This is needed for instance in order to get t-mwe-files annotated by multiple annotators, because the transformation can be run only on one s-file at a time. So to merge two s-files with one t-file, first one is merged statically, creating t-mwe-file and then this file is either merged dynamically by opening a different s-file while this t-mwe-file is present in the same directory, or by running this script again with a different s-file, resulting in t-mwe-file with both annotations.

`upgrade_st_data.pl` – Detect the format of s-files and if they are in the legacy format[8]used by `SemAnn` during annotations, correct the data.

[4] Developed by Petr Pajas

[5] The directory with `TrEd` extensions is platform dependent. On Linux and Mac OS X operating systems it is `~/.tred/extensions/`.

[6] The official documentation of extensions is a chapter of `TrEd` user manual: `http://ufal.mff.cuni.cz/~pajas/tred/ar01s17.html`.

[7] See Figure 4.5

[8] Our original PML schema for s-data was incorrect, in terms of PML specification. That is,

contrib.mac – The main (required) file for TrEd macros used in an extension. By convention it often just includes other files that really implement the macros. We keep the convention and all the macros are in display_mwe_groups.mak.

display_mwe_groups.mak – All our macros, since all that our extension really does, is highlighting the MWEs using TrEd node groups. Thes file also contains a slightly tricky function "detect", that detects whether the extension can handle the data being opened by TrEd.[9]

SDataMerge.pm – The core of the extension. This Perl module contains the functions to upgrade the legacy invalid s-data files to the valid ones, and the functions to merge the valid -sdata into the t-data files, creating t-mwe-data files that can be displayed and/or searched in TrEd.

pmlbackend-conf.inc – Alows to open unsupported files (st-files) and transform them on the input. See the Listing 4.2.

sdata_schema.xml – PML schema for the input s-files, described in detail in Section 4.3. It is used in the tdata_mwe_schema.xml, see below.

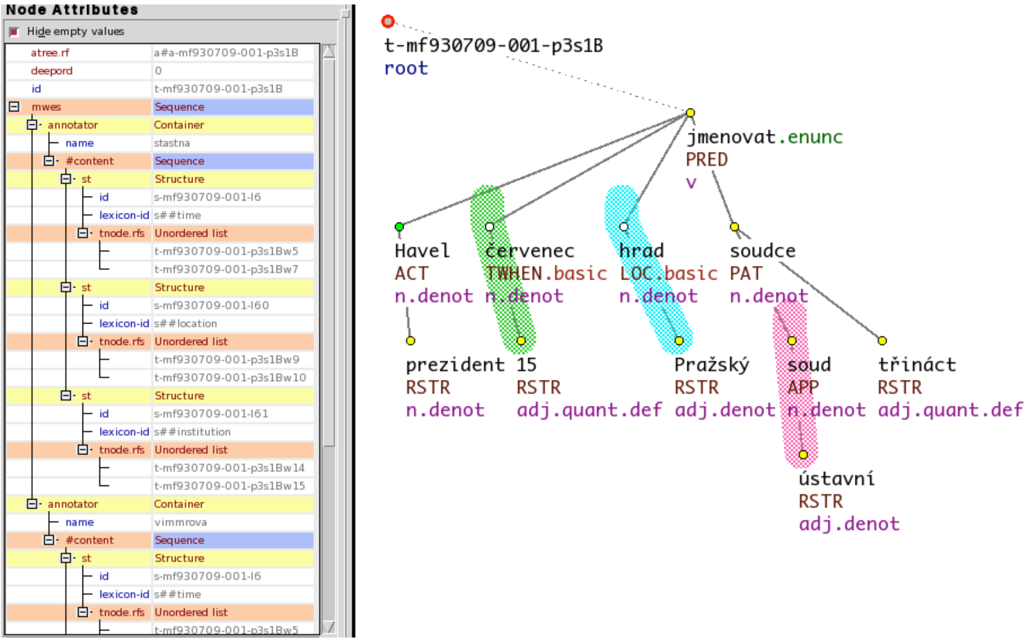
tdata_mwe_schema.xml – The PML schema of the t-data enhanced with information from the s-files. It imports the complete t-data schema, then imports st-node.type from s-data schema, and using these defines a new structured attribute of t-root (a root node of a tectogrammatical tree): The t-root can have an attribute 'mwes' to contain any MWEs. That attribute must have at least one child 'annotator' with an attribute 'name' that stores the annotator's name, and a content, that is a sequence (i.e. list) od st-nodes. See Figure 4.4 for an illustration.

Listing 4.2: pmlbackend-conf.inc.
Written by Petr Pajas – it allows an extension to use a Perl transformation on the input file that is not directly supported by any existing backend. In the commented section we can see that also any arbitrary shell command outputting valid PML to STDOUT can be used as an alternative transformation.

```
<?xml version=”1.0” encoding=”utf-8”?>
<pmlbackend
  xmlns=”http://ufal.mff.cuni.cz/pdt/pml/”
```

however, the form of data we used during the whole course of annotations and consequently all the legacy scripts expect this form.

[9] The tricky part is, what happens when several extensions claim the data. Deciding, which extension is the "right one" is not always trivial.



Node Attributes

■ Hide empty values

atree.rf	a#a-mf930709-001-p3s1B
deepord	0
id	t-mf930709-001-p3s1B
⊟ mwes	Sequence
⊟ annotator	Container
name	stastna
⊟ #content	Sequence
⊟ st	Structure
id	s-mf930709-001-l6
lexicon-id	s##time
⊟ tnode.rfs	Unordered list
	t-mf930709-001-p3s1Bw5
	t-mf930709-001-p3s1Bw7
⊟ st	Structure
id	s-mf930709-001-l60
lexicon-id	s##location
⊟ tnode.rfs	Unordered list
	t-mf930709-001-p3s1Bw9
	t-mf930709-001-p3s1Bw10
⊟ st	Structure
id	s-mf930709-001-l61
lexicon-id	s##institution
⊟ tnode.rfs	Unordered list
	t-mf930709-001-p3s1Bw14
	t-mf930709-001-p3s1Bw15
⊟ annotator	Container
name	vimmrova
⊟ #content	Sequence
⊟ st	Structure
id	s-mf930709-001-l6
lexicon-id	s##time
⊟ tnode.rfs	Unordered list
	t-mf930709-001-p3s1Bw5

Figure 4.4: A tectogrammatical tree with a `mwes` attribute containing the annotations of two annotators and a corresponding tree with visualisation of this annotation.

```
xmlns:xi=”http://www.w3.org/2001/XInclude”>
<head>
  <schema href=”pmlbackend_conf_schema.xml”/>
</head>
<transform_map>
    <transform id=”sdata” root=”sdata” ns=”http://ufal.mff.cuni.cz/pdt/pml/”>
      <in type=”perl” command=”require SDataMerge; return SDataMerge::transform(@_);”/>
    </transform>
    <!--

    other possiblity

    <transform id=”sdata2” root=”sdata” ns=”http://ufal.mff.cuni.cz/pdt/pml/”>
      <in type=”shell” command=”merge-s-and-t-layer.pl”>
        <param name=”-S”></param>
```

```
          </in>
        </transform>

        −−>
    </transform_map>
</pmlbackend>
```

Listing 4.3: tdata_mwe_schema.xml.

```
<?xml version=”1.0” encoding=”utf−8”?>
<pml_schema xmlns=”http://ufal.mff.cuni.cz/pdt/pml/schema/” version=”1.1”>
  <revision>0.1</revision>
  <description> PDT 2.0 tectogrammatic trees with multiword lexemes and named entities </description>
  <reference name=”adata” readas=”trees”/>
  <import schema=”tdata_schema.xml”/>
  <import type=”st−node.type” schema=”sdata_schema.xml”/>
  <derive type=”st−node.type”>
      <structure name=”s−node”>
        <member as_attribute=”1” name=”id” required=”1”><cdata format=”ID”/></member>
      </structure>
  </derive>
  <derive type=”t−root.type”>
    <structure role=”#NODE” name=”t−root”>
      <member name=”mwes” required=”0”>
        <sequence content_pattern=”(annotator)+”>
          <element name=”annotator”>
            <container>
              <attribute name=”name” required=”1”>
                <cdata format=”any”/>
              </attribute>
              <sequence>
                <element name=”st” type=”st−node.type”/>
```

```
                            </sequence>
                        </container>
                    </element>
                </sequence>
            </member>
        </structure>
    </derive>
</pml_schema>
```

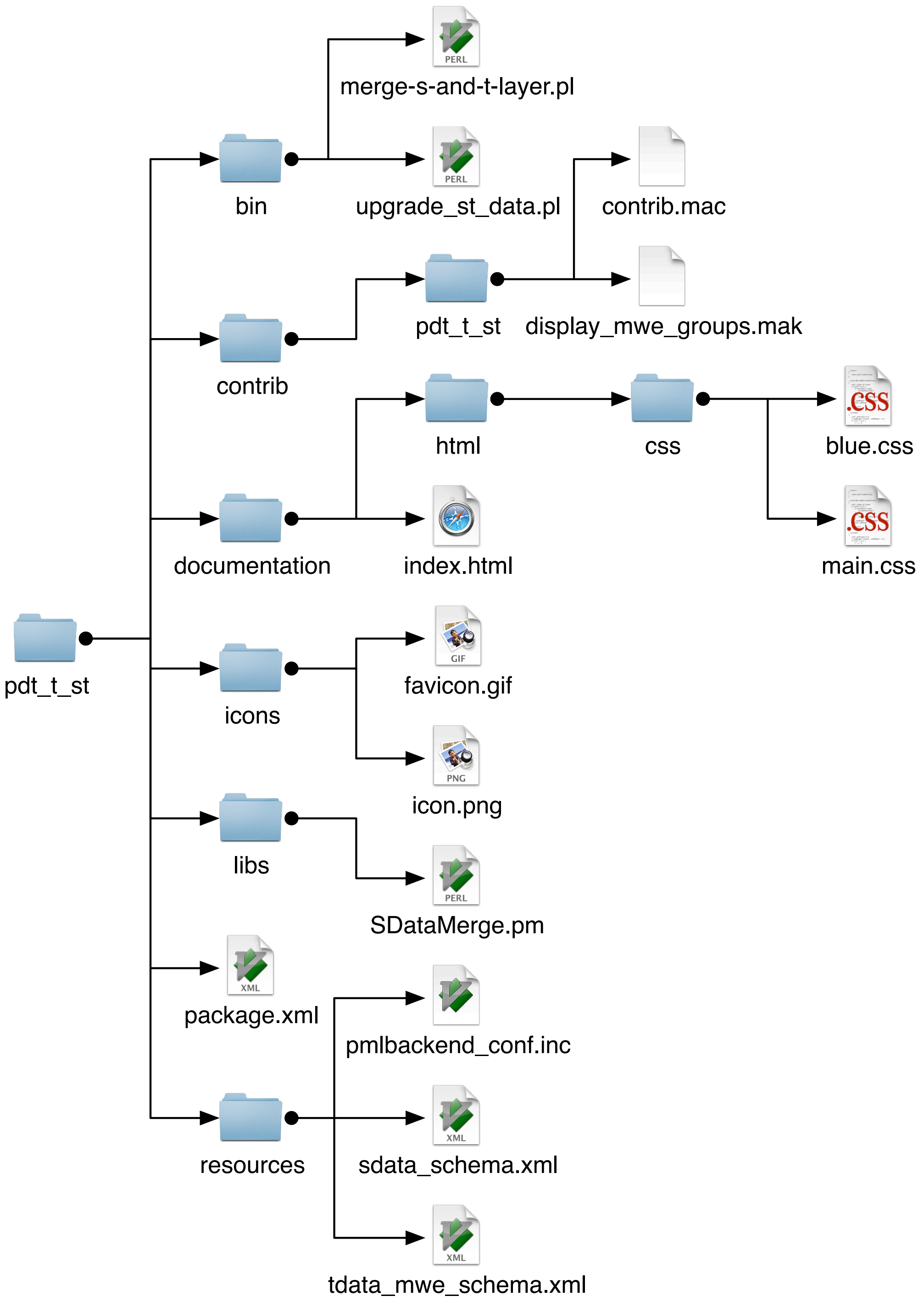


Figure 4.5: The structure of the 'pdt-t-st' TrEd extension.

Listing 4.4: s-data PML schema

```
<?xml version=”1.0” encoding=”utf−8”?>
<pml_schema xmlns=”http://ufal.mff.cuni.cz/pdt/pml/schema/” version=”1.1”>
 <description>PDT 2.0 sense (WSD) annotation</description>
 <reference name=”mdata” readas=”dom”/>
 <reference name=”adata” readas=”dom”/>
 <reference name=”tdata” readas=”dom”/>
 <reference name=”semlex”/> <!−− for SemLex as well as Vallex, etc. −−>

 <root name=”sdata”>
   <structure>
     <member name=”meta” required=”1”>
       <structure>
         <member name=”annotation_info”>
           <structure name=”s−annotation−info”>
             <member name=”lexicon” required=”1”><cdata format=”any”/></member>
             <member name=”annotator” required=”1”><cdata format=”any”/></member>
             <member name=”version_info”><cdata format=”any”/></member>
             <member name=”desc”><cdata format=”any”/></member>
           </structure>
         </member>
       </structure>
     </member>
     <member name=”wsd” required=”1”>
       <!−− No mixing of references to different layers within one file −−>
       <sequence content_pattern=”((sm)+|(sa)+|(st)+)”>
         <element name=”sm” type=”sm−node.type”/>
         <element name=”sa” type=”sa−node.type”/>
         <element name=”st” type=”st−node.type”/>
       </sequence>
     </member>
```

```
  </structure>
</root>

<!-- sense node, a type with elements common for all sense nodes -->
<!-- s-node id is constructed as a concatenation of a file ID and the number
   denoting a lexia instance in the given file.-->
<type name="s-node.type">
  <structure name="s-node">
    <member as_attribute="1" name="id" role="#ID" required="1"><cdata format="ID"/></member>
    <member as_attribute="1" name="src" required="0"><cdata format="any"/></member>
    <member name="lexicon-id" required="1"><cdata format="any"/></member>
  </structure>
</type>

<!-- s-node linking a sense to a set of m-nodes -->
<!-- Mostly used for old annotation of files that don't have t-layer -->
<derive name="sm-node.type" type="s-node.type">
  <structure name="sm-node">
    <member name="mnode.rfs">
      <list ordered="0">
        <cdata format="PMLREF"/>
      </list>
    </member>
  </structure>
</derive>

<!-- s-node linking a sense to a set of a-nodes -->
<derive name="sa-node.type" type="s-node.type">
  <structure name="sa-node">
    <member name="anode.rfs">
      <list ordered="0">
        <cdata format="PMLREF"/>
      </list>
    </member>
  </structure>
  </derive>

  <!-- s-node linking a sense to a set of t-nodes -->
  <derive name=”st-node.type” type=”s-node.type”>
    <structure name=”st-node”>
      <member name=”tnode.rfs”>
        <list ordered=”0”>
          <cdata format=”PMLREF”/>
        </list>
      </member>
    </structure>
  </derive>

</pml_schema>
```

Listing 4.5: An s-data file. The annotation includes named entities (identifiable by their special `SemLex` IDs) as welll as other MWEs and also an automatically pre-annotated MWE (line 19)

```
1  <?xml version=”1.0” encoding=”utf-8”?>
2  <sdata xmlns=”http://ufal.mff.cuni.cz/pdt/pml/”>
3    <head>
4      <schema href=”sdata_schema.xml”/>
5      <references>
6        <reffile id=”t” name=”tdata” href=”ln94203_3.t.
            gz”/>
7        <reffile id=”s” name=”semlex” href=”semlex.xml
            ”/>
8      </references>
9    </head>
10   <meta>
11     <annotation_info>
12       <lexicon>SemLex Devel</lexicon>
13       <annotator>vimmrova</annotator>
14       <version_info/>
15       <desc/>
16     </annotation_info>
17   </meta>
18   <wsd>
19     <st id=”s-ln94203-3-12” src=”auto”>
20       <lexicon-id>s##person</lexicon-id>
21       <tnode.rfs>
22         <LM>t#t-ln94203-3-p4s2w12</LM>
23         <LM>t#t-ln94203-3-p4s2w13</LM>
24       </tnode.rfs>
25     </st>
26     <st id=”s-ln94203-3-13”>
27       <lexicon-id>s##institution</lexicon-id>
28       <tnode.rfs>
29         <LM>t#t-ln94203-3-p4s2w15</LM>
30         <LM>t#t-ln94203-3-p4s2w17</LM>
31         <LM>t#t-ln94203-3-p4s2w18</LM>
32         <LM>t#t-ln94203-3-p4s2w19</LM>
33         <LM>t#t-ln94203-3-p4s2w20</LM>
34         <LM>t#t-ln94203-3-p4s2w21</LM>
35       </tnode.rfs>
```

```
</st>
<st id="s-ln94203-3-14">
  <lexicon-id>s##location</lexicon-id>
  <tnode.rfs>
    <LM>t#t-ln94203-3-p5s1w7</LM>
    <LM>t#t-ln94203-3-p5s1w8</LM>
  </tnode.rfs>
</st>
<st id="s-ln94203-3-15">
  <lexicon-id>s##person</lexicon-id>
  <tnode.rfs>
    <LM>t#t-ln94203-3-p5s1w19</LM>
    <LM>t#t-ln94203-3-p5s1w21</LM>
  </tnode.rfs>
</st>
<st id="s-ln94203-3-16">
  <lexicon-id>s##location</lexicon-id>
  <tnode.rfs>
    <LM>t#t-ln94203-3-p5s6w2</LM>
    <LM>t#t-ln94203-3-p5s6w3</LM>
  </tnode.rfs>
</st>
<st id="s-ln94203-3-17">
  <lexicon-id>s#0000025022</lexicon-id>
  <tnode.rfs>
    <LM>t#t-ln94203-3-p2s1w17</LM>
    <LM>t#t-ln94203-3-p2s1w18</LM>
  </tnode.rfs>
</st>
<st id="s-ln94203-3-18">
  <lexicon-id>s#0000010260</lexicon-id>
  <tnode.rfs>
    <LM>t#t-ln94203-3-p3s2w20</LM>
    <LM>t#t-ln94203-3-p3s2w21</LM>
    <LM>t#t-ln94203-3-p3s2w22</LM>
  </tnode.rfs>
</st>
<st id="s-ln94203-3-19" src="auto">
  <lexicon-id>s#0000010260</lexicon-id>
  <tnode.rfs>
    <LM>t#t-ln94203-3-p4s3w7</LM>
```

```
        <LM>t#t-ln94203-3-p4s3w8</LM>
        <LM>t#t-ln94203-3-p4s3w9</LM>
      </tnode.rfs>
    </st>
    <st id="s-ln94203-3-l10" src="auto">
      <lexicon-id>s#0000010260</lexicon-id>
      <tnode.rfs>
        <LM>t#t-ln94203-3-p5s6w14</LM>
        <LM>t#t-ln94203-3-p5s6w15</LM>
        <LM>t#t-ln94203-3-p5s6w16</LM>
      </tnode.rfs>
    </st>
    <st id="s-ln94203-3-l11">
      <lexicon-id>s#0000031685</lexicon-id>
      <tnode.rfs>
        <LM>t#t-ln94203-3-p5s5w7</LM>
        <LM>t#t-ln94203-3-p5s5w8</LM>
        <LM>t#t-ln94203-3-p5s5w9</LM>
      </tnode.rfs>
    </st>
  </wsd>
</sdata>
```

Chapter 5

SemAnn

The annotation tool SemAnn is written in Perl 5[1] with Perl/Tk[2] 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

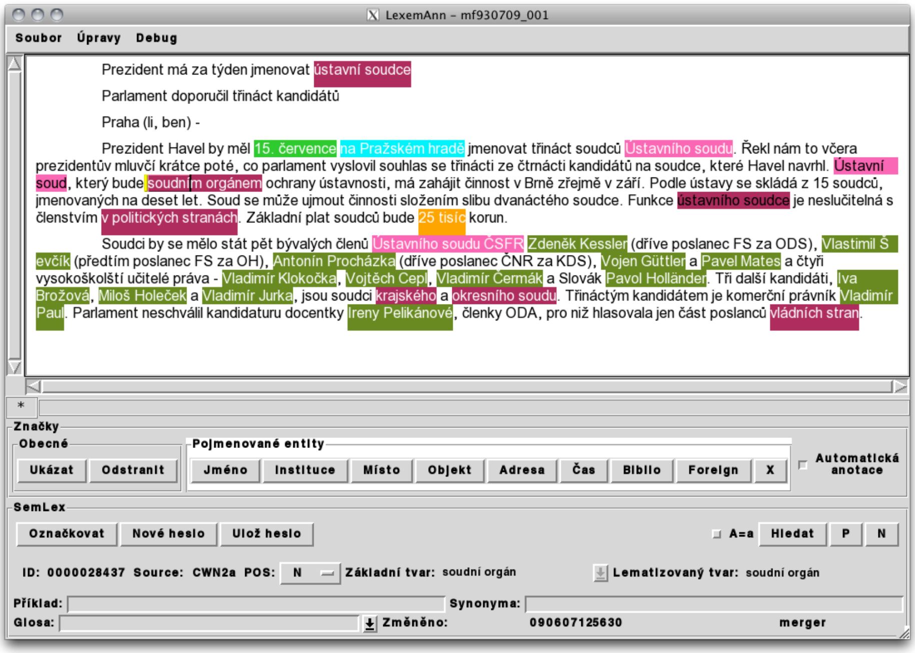
  - generate surface sentences from tectogrammatical trees in t-files that are then displayed in the SemAnn GUI,
  - perform all the on-the fly pre-annotations (Section 7.3)

  A TrEd engine without a GUI with a few modifications aimed towards batch processing is called b-TrEd. n-TrEd is essentially a modification of b-TrEd that allows it to run as a daemon and process scripts over a network. We opted for n-TrEd, even though we ran it locally, because it can run as a daemon, thus eliminating a significant startup penalty of b-TrEd. What we call "n-TrEd backend" is thus the running n-TrEd instance itself (that is started by the SemAnn during start-up, if there has not been a running instance detected), and the scripts used to generate the sentences that are displayed in SemAnn and to pre-annotate MWEs using their tectogrammatical tree structures.[3]

[1] www.perl.org; dev.perl.org/perl5

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

[3] these scripts were written entirely by E. Bejček (2010).



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.

□ Automatická anotace

□ A=a

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

Příklad: Synonyma:

Glosa: Změněno: 090607125630 merger

Figure 5.1: 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.pm is used to read, save, query, and edit SemLex.
- 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.

5.1 User interface

The user interface (shown in Figure 5.1) is divided three main parts: The text widget displaying the annotated text, a row of buttons to create annotations by NE tags, show info on annotations, or remove tags, and an editor of SemLex.

5.1.1 Text widget

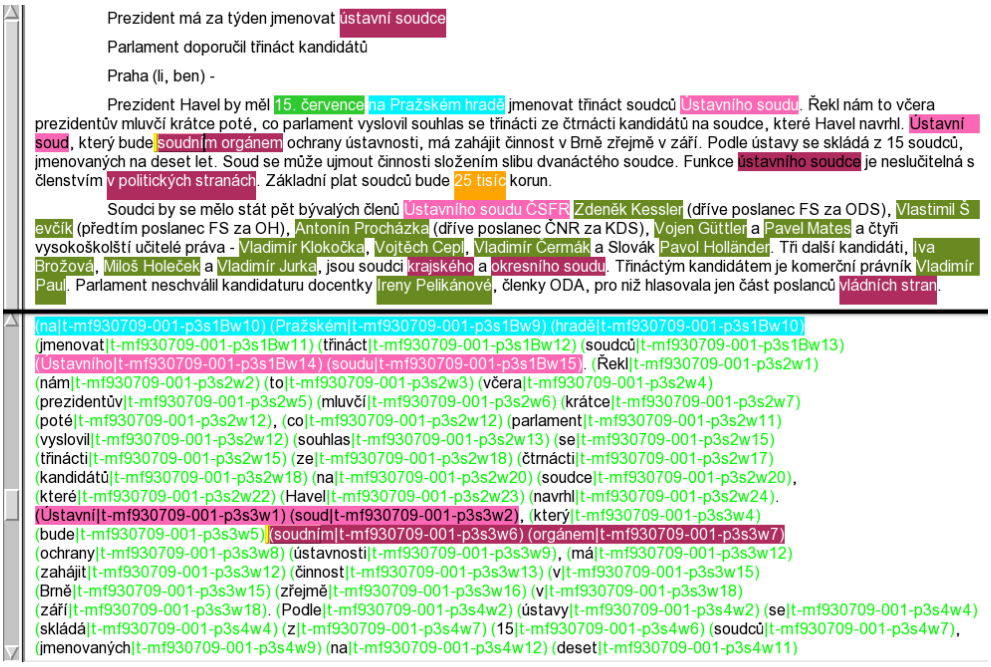
The text, displayed for the annotator, is generated from the tectogrammatical trees (using also information from lower layers). That is, why for each document to be annotated, all the PDT files must be present (t-, a-, m-, and w-layer).

It is possible to generate the surface (plain text) sentence from a t-tree using the 'built-in' function PML_T::GetSentenceString($root). Such a sentence is complete, correct, has correct spacing around punctuation, but it contains no relation to the t-layer anymore. And we want to keep this connection in order to be able to annotate the t-nodes, not just words (see Section 7). Thus Eduard Bejček wrote the script get_sent_t-layer.btred, that creates a representation connecting words in a sentence with tectogrammatical IDs of the t-nodes from which these words are generated. This representation is actually input into the text widget and everything but the words is hidden from the annotators' view. *The tecto-IDs are then what gets really annotated,* using *the words.* The full representation can, however, be displayed using Debug menu commands. It is shown in Figure 5.2 in comparison to the "plain text" as normally displayed (with no actual annotations to keep the view simple).

5.1.2 Annotation buttons

The row of buttons below the text widget and the status bar is rather straightforward:

The first group (from the left) contains two buttons that are connected to general commands used for all annotations (NEs and other SemLex entries alike). The



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.

```
(na|t-mf930709-001-p3s1Bw10) (Pražském|t-mf930709-001-p3s1Bw9) (hradě|t-mf930709-001-p3s1Bw10)
(jmenovat|t-mf930709-001-p3s1Bw11) (třináct|t-mf930709-001-p3s1Bw12) (soudců|t-mf930709-001-p3s1Bw13)
(Ústavního|t-mf930709-001-p3s1Bw14) (soudu|t-mf930709-001-p3s1Bw15). (Řekl|t-mf930709-001-p3s2w1)
(nám|t-mf930709-001-p3s2w2) (to|t-mf930709-001-p3s2w3) (včera|t-mf930709-001-p3s2w4)
(prezidentův|t-mf930709-001-p3s2w5) (mluvčí|t-mf930709-001-p3s2w6) (krátce|t-mf930709-001-p3s2w7)
(poté|t-mf930709-001-p3s2w12), (co|t-mf930709-001-p3s2w12) (parlament|t-mf930709-001-p3s2w11)
(vyslovil|t-mf930709-001-p3s2w12) (souhlas|t-mf930709-001-p3s2w13) (se|t-mf930709-001-p3s2w15)
(třinácti|t-mf930709-001-p3s2w15) (ze|t-mf930709-001-p3s2w18) (čtrnácti|t-mf930709-001-p3s2w17)
(kandidátů|t-mf930709-001-p3s2w18) (na|t-mf930709-001-p3s2w20) (soudce|t-mf930709-001-p3s2w20),
(které|t-mf930709-001-p3s2w22) (Havel|t-mf930709-001-p3s2w23) (navrhl|t-mf930709-001-p3s2w24).
(Ústavní|t-mf930709-001-p3s3w1) (soud|t-mf930709-001-p3s3w2), (který|t-mf930709-001-p3s3w4)
(bude|t-mf930709-001-p3s3w5) (soudním|t-mf930709-001-p3s3w6) (orgánem|t-mf930709-001-p3s3w7)
(ochrany|t-mf930709-001-p3s3w8) (ústavnosti|t-mf930709-001-p3s3w9), (má|t-mf930709-001-p3s3w12)
(zahájit|t-mf930709-001-p3s3w12) (činnost|t-mf930709-001-p3s3w13) (v|t-mf930709-001-p3s3w15)
(Brně|t-mf930709-001-p3s3w15) (zřejmě|t-mf930709-001-p3s3w16) (v|t-mf930709-001-p3s3w18)
(září|t-mf930709-001-p3s3w18). (Podle|t-mf930709-001-p3s4w2) (ústavy|t-mf930709-001-p3s4w2) (se|t-mf930709-001-p3s4w4)
(skládá|t-mf930709-001-p3s4w4) (z|t-mf930709-001-p3s4w7) (15|t-mf930709-001-p3s4w6) (soudců|t-mf930709-001-p3s4w7),
(jmenovaných|t-mf930709-001-p3s4w9) (na|t-mf930709-001-p3s4w12) (deset|t-mf930709-001-p3s4w11)
```

Figure 5.2: Comparison of the plain text, as normally displayed, and the underlying representation that is used to relate the actions of annotators who mark words, to the references to tectogrammatical nodes that are actually saved in the annotation files.

first button shows a tag (i.e. the corresponding `SemLex` entry), on the word in focus (the word that is selected, or in which the cursor is placed). The second button removes the tag on the word in focus.

The second group of buttons simply creates NE tags over the selection.

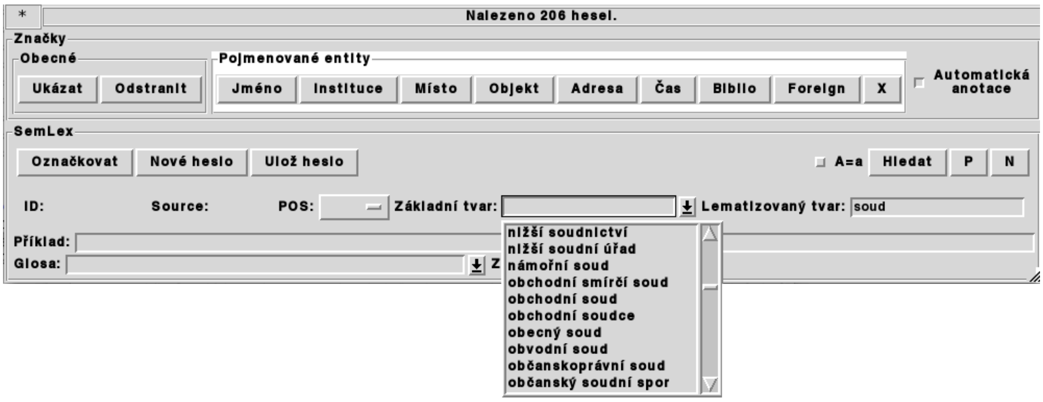
The last check button toggles the on-the-fly pre-annotation of other instances of the same MWE that annotator annotates, in the rest of the text (pre-annotation type D, see p. 59).

5.1.3 SemLex editor

The `SemLex` editor and browser (see the lower part of Figure 5.1) simply displays SemLex entries, allows to edit them, or to search SemLex by basic or lemmatised forms (see Figure 5.3), and browse the search results (using their basic forms). There is also a function to annotate the selected words (t-nodes) with the current SemLex entry. It is mapped simply to the return/enter key, once the focus moves from the SemLex part of the GUI to the text widget.

The attributes of an entry that are displayed include the basic and lemmatised form of an entry, its ID and source, example of usage, synonyms, if present, and a gloss. There is also a time stamp and a signature of the last modification. The attributes of a SemLex entry are explained in detail in Section 6.3.1.

The search string is by default matched as a substring, and there is a check box to toggle case sensitivity. However when needed (and in case an annotator has the knowledge), full Perl regular expressions can be used.



Nalezeno 206 hesel.

Lematizovaný tvar: soud

nižší soudnictví
nižší soudní úřad
námořní soud
obchodní smírčí soud
obchodní soud
obchodní soudce
obecný soud
obvodní soud
občanskoprávní soud
občanský soudní spor

Figure 5.3: A result of a search in SemLex by a substring of the lemmatised form: 206 entries were found (see the status line at the top). Browsing the basic forms of the results.

5.1.4 General UI remarks

Inspired by command mode of modal text editors and by some annotation regimes of TrEd, we made all the annotation commands single-letter. That was made possible by making the text read-only. Since the letters are not used for input, they can be mapped to commands. All the command (and so the buttons) are named (in Czech) in such manner, that their first letter can be mapped to perform the command. Only the command for removing annotation is mapped to the capital letter ('O' for 'odstranit') for safety reasons.

5.2 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.

During several previous annotation projects (cf. Hajič et al., 2004; Bejček et al., 2006) we got repeatedly into situations when it was very how do the annotators exactly work, and sometimes even how much they work. The way they work is, however, crucial in both estimating a fair price of their work, and also in estimating the correct way annotations should proceed. For information on how we made preliminary estimation of the speed of annotations and some directions for future work see Section 7.5.

The log files are nevertheless useful also directly to annotators during their daily work. They provide information necessary for persistent undo and redo. Even when the file has been partially annotated long time ago, an annotator can review the last steps taken and continue with more information.

Chapter 6

SemLex

6.1 Named entities

As we have already stated, we use a modified version of NE classification by Ševčíková et al. (2007) Our types of entities are:

1. “a name of a person or an animal”,
2. “institution”,
3. “location”,
4. “other object” (used for names of books, units of measurement, biological names of plants and animals),
5. “address”,
6. “time”,
7. “bibliographic entry”,
8. “foreign expression” and
9. “other entity”

Compared to the original, the classification is altered because we do not use embedded entities. In the original, Ševčíková et al. use a bracketing approach, in which entities of all types are further structured into smaller parts. We also altered some rules for classifying particular types of entities as follows:

- We do not distinguish *names of animals* as a distinct type. Animal names are considered the same type as the names of persons.

- We also merge the *names of media* (newspapers, TV stations, …) with names of companies. The reasons are mainly: a) it seems arbitrary to distinguish specifically names of media. b) It is also often hard to distinguish whether a name is a name of media or a company that owns or runs the media. At the same time there is usually little reason to try to make this distinction.

- *Numbers with non-quantifying meaning* are merged with addresses. Since the subtypes defined for this type are `zip code`, `street number` and `phone/fax number`, this merge is quite natural, especially since these occur mostly as parts of addresses and we do not annotate any embedded entities.

Some frequent names of persons, institutions or other objects (e.g. film titles) are being added into SemLex during annotation (while keeping the information about their NE type), because this allows for their following occurrences to be pre-annotated automatically (see Section 7.3). For others, such as common addresses or bibliographic entries, it makes but little sense, because they most probably will not reappear during the annotation.

6.2 MWEs from other dictionaries

The base of `SemLex` has been composed of MWEs extracted from Czech WordNet (Smrž, 2003), Eurovoc (eur, 2007) and Dictionary of Czech Phraseology and Idiomatics (SCFI, Čermák et al., 1994).

For an explanation of a special use of a SCFI subset see Point A in Section 7.3.

The entries added by annotators must have a gloss. Annotators define it informally but as well as possible and we extract an example of usage and the basic form from the annotation automatically. The meaning information in a gloss can be revised by a lexicographer, based on annotated occurrences.

Most of the MWEs (or potential MWEs, in some cases) that were complied from the resources described in this section, were actually never used during annotation. Nevertheless, it was very beneficial to have an annotation lexicon to start with.

6.2.1 Eurovoc

We extracted a huge amount of multiword entries from the Eurovoc thesaurus (eur, 2007). Many of them were, however, not MWEs. They were often phrases, sometimes whole clauses. After some filtering we ended up with 15,176 potential MWEs that were added to SemLex.

6.2.2 Czech WordNet

We used the Czech WordNet (Smrž, 2003; Pala and Smrž, 2004) in its most up-to-date development version available at the time. It was significantly larger than then-current public release. It was however also not yet the more recent version described by Pala and Hlaváčková (2007). We had used this version of CWN previously in a different annotation project (Bejček et al., 2006).

A significant benefit of CWN is its simple structure and XML implementation. Since the basic unit of any WordNet is a *synset* – a set of synonyms, we took advantage of this and created an entry for each synonym of a synset, storing the other synonyms in an appropriate attribute of a SemLex entry. These synonyms proved quite valuable. Since most CWN entries have no glosses, they also serve as a definition.

We acquired 11,345 MWEs from CWN and stored them in our preliminary SemLex.

6.2.3 Dictionary of Czech Phraseology and Idiomatics

By processing an electronic version of Dictionary of Czech Phraseology and Idiomatics (SCFI) (Čermák et al., 1994) we created *3,985 initial MWEs as entries in SemLex.*

SCFI has several nice properties. First of all, its entries contain a lot of information, especially compared to the other two resources we used. Rich explanatory glosses that even contain examples are one thing worth mentioning. Translation equivalents in English, German, French and Russian are another very valuable property, especially considering the dictionary contains large amount of idioms.

The slightly problematic property of the source data we obtained was a proprietary 8bit text format that essentially copies the paper version of a dictionary, thus not giving precise machine-readable distinctions of properties of entries.

6.3 Structure of `SemLex`

From the technical point of view, `SemLex` is a simple list of entries. It is stored in YAML format, which makes it easily readable in its source form using any Unicode-aware text editor. Since YAML is a data serialization format and not a markup language, it can also store information needed to represent data as objects in the Perl OO model.

In addition to the `SemLex` itself we also build an index of basic forms on some special occasions. It is not needed during annotation, so we do not create the index normally but it can be called by passing a second parameter[1] to the function `SemLex::load_yaml()`.

[1] The parameter is evaluated as a BOOL type.

6.3.1 SemLex entry

An entry is composed of several user-editable attributes, and some read-only and machine-generated metadata:[2]

```
- !!perl/hash:SemLex_heslo
  BASIC_FORM: vysoké kruhy
  CREATED: 070115163056
  EXAMPLE: ’’
  GLOSS: ’’
  ID: 0000017495
  LEMMATIZED: vysoký kruh
  MODIFIED: 090607124212
  MODIFIER: merger
  MORPHO_TAGS: AAIP2----3A---- NNIP2-----A----
  ORIGID: ’’
  POS: ’’
  SOURCE: SCFI
  SYNONYMS: []
  TREE_STRUCT:
    -
      - kruh
      - ~
    -
      - vysoký
      - 0
```

- !!perl/hash:SemLex_heslo – The first line of a SemLex entry provides a lot of information. We describe its parts from the left to the right:

- The hyphen at the beginning of the line, together with indentation of the lines that follow, indicates that this is an array element in YAML. Remember that we said that SemLex is a simple list of its entries. Thus we chose to implement is with an array, as it is both sufficient and very efficient.

!!perl This string says that the array element actually represents a serialisation of a Perl object.

/hash The object is implemented as a hash.

:Semlex_heslo The object belongs to the class Semlex_heslo.

[2] By “user-editable” we mean editable in SemAnn. Everything is of course editable in the YAML source format using a text editor, but that is not what a typical user that we have in mind does.

BASIC_FORM – The basic form of a MWE. We could call it a "lemma" of a MWE, but we do not find it suitable for several reasons:

- In many languages including Czech it often contains word forms in other than the basic form for the given word on its own. I.e. "vysoké učení" contains a feminine suffix of the adjective "vysoký" (high) because of the required agreement in gender with the noun, whereas the traditional lemma of adjectives in Czech is in the masculine form.
- It could be confused with the attribute "lemmatized" that means something completely different.

EXAMPLE – An example sentence or collocation that illustrates the prevalent use of the MWE

GLOSS – Primarily used for an explanation of the MWE, much like in traditional dictionaries.

Secondary use of this attribute is for additional notes or processing instructions. Annotators put specially formated notes into this field to mark that the entry has some special property that we do not have an attribute for, that the entry should be removed, or they could indicate that they have just some other note for the entry. This secondary use of GLOSS and the type of note was marked by special format of the beginning of the note:

`***(<type>)` – The entry is a NE that the user wants to add into `SemLex`. The usual reason is that the NE is common in the annotated text. When it is added into `SemLex`, it can be pre-annotated automatically more efficiently (see Section 7.3 for details). `<type>` means that at this place there is a name of one of the types of NEs as described in Section 6.1.[3]

`***derived from: <ID>` – The entry is derived from another entry that already exists in `SemLex`.

`***remove` – The entry is not a MWE, thus its instances should not be annotated and the entry should be removed from `SemLex`.

`***<anything else>` – Other notes. The entry must be inspected and something (other than removing it) must be done.

LEMMATIZED – "Lemmatised `BASIC_FORM`", i.e. take the basic form of an entry and substitute each form with its morphological lemma. This attribute is used for pre-annotation of entries that have not been annotated yet, so their tree structure has yet to be identified. For more details see Section 7.3 on page 58.

[3] The notes actually use Czech names of these types.

MODIFIER – For newly created entries, this attribute is empty regardless whether they were created by modifying entries from other dictionaries during original creation of SemLex, or whether they were created by annotators during annotation. Is is used to mark that the entry was modified after its creation and who last modified the entry. See 6.3.1

MORPHO_TAGS – Morphological tags corresponding to the `BASIC_FORM`. The tags were acquired automatically by running the morphological tagger of Jan Hajič (2004). This is a suplementary information that has not been used during later stages of annotation. It was implemented for pre-annotation using only morphological layer or alternativelly only plain text and morphological tagger. After some initial testing we have not used this pre-annotation, so it does not appear in Section 7.3 and it is not implemented in current `SemAnn` workflow.

The morphological tags could however still prove useful if `SemLex` should be used for annotatiuon of resources without tectogrammatical layer. Then it could be useful to employ this type of simpler pre-annotation.

ORIGID – If the entry comes from some other existing dictionary, this is the original ID of the entry in that source dictionary (identified by the attribute `SOURCE`).

POS – The part of speech of the entry as a unit. Usually it corresponds to the part of speech of the syntactic head of the entry in terms of the underlying tectogrammatic tree structure, but there are some exceptions:

N – noun;

A – adjective; *trvale udržitelný, ekonomicky aktivní, do očí bijící*

V – verb; *působit jako blesk z čistého nebe , zaujmout stanovisko, mít dohled, spadl (komu) kámen ze srdce*

D – adverb; e.g. *mezi čtyřma očima, na lavičce*[4], or *o dům dál*

I – interjection;

F – foreign; often Latin, Greek, but also other idioms that are already "native" to the Czech language enough to include them in a dictionary instead of annotating them just as a foreign entity, e.g. *hip hop, a la, de iure*.

[4] Trenér Borovička se zatím nerozhodl, zda král střelců Siegl začne v základní sestavě, nebo zase jen na lavičce.

N/A – not applicable; used for proverbs, sayings, other idioms forming whole sentences, or idioms with unclear part of speech, e.g. *stručně řečeno; stal se kozel zahradníkem, Tady je dobrá rada drahá.*

The entries in the original SČFI do not carry the information on the part of speech, so we were not able to fill the attribute in the beginning. Thus the entries from SČFI have the POS information only in case they were manually edited. In that case the POS is always added, because the `SemAnn` interface does not allow the edited entry to be saved without indicating the part of speech.[5]

SOURCE – Where did the entry come from. Possible values are `CWN2a`, `Eurovoc`, `SCFI`, or `<annotator>`. `<annotator>` means that the entry was created during annotations and stands for an identifier of the annotator who created it. An annotator found a MWE, searched the `SemLex` for the expression, and decided a new entry is needed. When the entry is created, the annotator's identifier is written as the SOURCE. This attribute is used to trace the origin of entries during analysis of annotations and when merging individual annotators' SemLexes.

When the entry is a result of merging several entries from different sources (either different source dictionaries or several annotators created the same entry), the value is a concatenation of the sources. In case that automatic merge was not possible `SOURCE`

SYNONYMS – A list of synonyms (implemented as an array). In case of Entries from WordNet, the synsets were split to individual entries, but the relation between them was kept via the ORIGID attribute and the basic forms of the synonymous MWEs were copied into this attribute as the synonyms of the MWE. This attribute helps annotators to understand an entry's meaning, especially since most synsets in the WordNet did not have any glosses or examples. See Section 6.2.2 for more information on entries derived from WordNet. Occasionally annotators added synonyms manually, if they just happened to think about them when using an entry. It was however by no means systematic work and it was not one of the goals of annotation.

TREE_STRUCT – The tree structure of an entry. It is implemented as an array (see above in the example entry). Each node in this tree structure has only two attributes: its tectogrammatical lemma, and an effective parent (identified by an index in the array).[6] This much simplified tectogrammatical repres-

[5] For details of the `SemAnn` user interface to `SemLex` see Section 5.1.

[6] '∼' means 'undef', i.e. the node is a root of the tree structure, because it does not have an effective parent. Indices start from zero, i.e. zero means the first element.

entation of an entry is sufficient for our purpose. It is a key to the most advanced pre-annotation that we employ: identification of future occurrences of the same tree structure in the text. This pre-annotation requires `n-TrEd`. See discussion of various types of pre-annotation in Section 7.3.

Chapter 7

Annotation

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 7.1 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 7.1: 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, as described in Section 6.3.1.

7.1 Hypothesis of a contiguous tectogrammatical structure of a MWE

Analysis of tree structures of MWEs in SemLex confirms our hypothesis. All the multiword expressions can be represented by a single contiguous tree structure. There is only one exception, but even that is more theoretical than practical problem.

When MWEs appear in coordinating constructions, e.g. *červené a bílé víno* (red and white wine), the common part of the expressions is not repeated. According to Mikulová et al. (2006) in this type of coordination the t-node common to both

MWEs should actually be duplicated, because these are clearly two different objects. The rule serves the purpose of distinguishing a coordination of two objects from a coordination of attributes of one object. From what we observed, however, the rule was not very well observed. There are many coordinations where the common part of two MWEs is represented by only one t-node, as in the example above.

Therefore we had no choice but to implement our tree structure of MWEs via "effective parent" relation, that skips coordinating conjunctions. That solves one problem well, but creates another one. There are MWEs that include coordinating conjunctions. And since these conjunctions are skipped when we construct tree structures of MWEs, resulting tree structures are contiguous with the exception of the conjunctions that remain unattached.

Pre-annotation of these MWEs still works, for the same reason that creates the unattached conjunctions: we use the effective parent relation, so we do not need the conjunctions.

7.2 Hypothesis of one tectogrammatical structure per MWE

Because MWEs tend to occur repeatedly in a text, we have decided to test pre-annotation both for speed improvement and for improving the consistency of annotations. We use an assumption that *all occurrences of a MWE share the same tectogrammatical tree structure*. We can call this assumption *"One tree per MWE,"* and view it as a modification of famous assumption "One sense per collocation" of Yarowsky (1993). In the surface representation of the MWE, there are *no restrictions on the word order* other than those imposed by the tree structure itself. Our assumption is however more inspired by an idea of Holub and Böhmová (2000), who proposed to use so called "Dependency microcontext structures (DMCS)" in information retrieval. DMCS were inspired by tectogrammatical tree structures, but modified a bit for easier extraction and handling. In contrast, we use tectogrammatical structures as they are.

771 `SemLex` entries have been used with more than one tectogrammatical tree structure in the annotated data.

Below we analyse several of the most important sources of these inconsistent t-trees (see Figure 7.1) and possible improvements:

- *Occasional lemmatisation errors.* They are not very frequent, but there is no efficient way to find and correct them before the annotations. So there is not much we can do but it is not very important. Our annotations can however serve as a source for automatic corrections.

```
ID 0000000636:
24x:    ---      2x:     ---     2x:     ---
-                -               -
  - zákoník        - zákon         - zákoník
  - ~              - ~             - ~


-                -               -
  - občanský       - občanský      - Občanský
  - 0              - 0             - 0
```

Figure 7.1: Three different tree structures in data for the SemLex entry 0000000636: "občanský zákoník".

- *Občanský* (Citizen (adj.)) in Figure 7.1 is a good example of bad lemmatisation. More typical cases of mixing literal and idiomatic meaning:
- *jižní Korea* vs. *Jižní Korea* (southern vs. South Korea),
- *rudé právo* vs. *Rudé právo* (red law vs. The Red Law, i.e. a Czechoslovak communist newspaper)

The occasional incorrectly lemmatised occurrence must be identified by annotators manually.

- *Annotator's mistake (not marking correct words).* When an annotator makes an error while marking a first occurrence of a MWE, the tree representation that gets stored in SemLex is incorrect. As a result, pre-annotation gives false positives or fails to work.

  It is therefore necessary to allow annotators to correct the tree structure of a SemLex entry, i.e. extend functionality of the annotation tool. Once all the types of pre-annotation are employed, this error can happen only once, because all the following occurrences of a MWE are pre-annotated automatically. We are currently working on these improvements.

- *Gender opposites, diminutives and augmentatives.* These are currently represented by variations of t-lemma. We believe that they should be represented by attributes of t-nodes that could be roughly equivalent to some of the lexical functions in the Meaning-text theory (see Mel'čuk (1996)). This should be tackled in some future version of PDT. Once resolved it would allow us to identify following (and many similar) cases automatically.

  - *obchodní ředitel* vs. *obchodní ředitelka*
    (lit.: managing director-man vs. m. director-woman)

  - *rodinný dům* vs. *rodinný domek*
    (lit.: family house vs. family little-house; but the diminutive *domek* means basically "family house")

  Currently we annotate these cases with the same MWE, but all the instances with the derived variants of t-lemma (like *ředitelka* or *domek* must be identified manually (see Section 7.3).

- *Newly established t-nodes corresponding to elided parts of MWEs in coordinations.* Since t-layer contains many newly established t-nodes, many of which cannot generate a surface word, our original decision was to hide all of these nodes from annotators and generate for them pure surface sentence. This decision resulted however in the current situation, when some MWEs in coordinations cannot be correctly annotated. For instance *První a druhá světová válka* (First and Second World War) is a coordination of two multiword lexemes. A tectogrammatical tree that includes it does have newly established t-nodes for "world" and "war" of the first lexeme but they are elided in the surface sentence.

Up to now we have not found any MWE such that its structure cannot be represented by a single tectogrammatical tree. 1.1% of all occurrences were not connected graphs, but this happened due to errors in data and due to our incorrect handling of coordinations with newly established t-nodes (see above). This corroborates our assumption that (disregarding errors) all occurrences of a MWE share the same tree structure. As a result, we started storing the tree structures in the SemLex entries and employ them in pre-annotation (D). This also allows us to use pre-annotation (C).

7.3 Pre-annotation

We employed four types of pre-annotation, only some of which are based on the assumption of unique tree structure (see Section 7.1):

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%.

7.4 Measuring the inter-annotator agreement

During the annotations we employed four annotators. Three of them annotated a significant amount of work, the fourth, who is not mentioned elsewhere in this text, helped with various experiments. The annotator identified by name sta replaced annotator sid, while vim worked with us during whole course of the project.

Below we give examples and describe parallel data of just one pair of annotators: (sid, vim). We saw little use in filling text with all the numbers for both pairs of annotators. In some places, where it seems meaningful, however, we report for instance the kappa scores fully.

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 7.2. Both annotators processed 1090 files in parallel. The data consists of 350,177 tokens representing 284,029 t-nodes.

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

7.4.1 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.

There are, however, a few sources of complications in measuring agreement of our task even by κ_w:

- Each tag of a MWE identifies a subtree of a tectogrammatical tree (represented on the surface by a set of marked words). This allows for partial agreement of tags at the beginning, at the end, but also in the middle of a surface interval (in a sentence). Instead, standard measures like κ assumes fixed, bounded items, which are assigned some categories.

- There is no clear upper bound as to how many (and how long) MWEs there are in texts. Cohen's κ_w counts agreement on known items and these are the same for both annotators. On the other hand, we want to somehow count agreement on the fact, that given word is not a part of MWE.

- There is not a clear and simple way to estimate the amount of agreement by chance, because it must include the partial agreements mentioned above.

Since we want to keep our agreement calculation as simple as possible but we also need to take into account the issues above, we have decided (as mentioned above) to start from κ_w as defined in Artstein and Poesio (2007): $\kappa_w = 1 - \frac{D_o}{D_e} = \frac{A_o - A_e}{1 - A_e}$ (explanation in Equation 7.2) and to make a few adjustments to allow for an agreement on non-annotation and an estimated upper bound. We explain these adjustments in following paragraphs on the example of paralled data of annotators sid and vim[1] and we summary quantitative data for this pair in Table 7.3.

Because we do not know how many MWEs there are in our texts, we need to *calculate the agreement over all t-nodes*, rather than just the t-nodes that "should be annotated". This also means that the theoretical maximal agreement (upper bound) U cannot be 1. If it was 1, it would be saying that all nodes are part of MWEs.

Since we know that $U < 1$ but we do not know its exact value, we use the *estimated upper bound* $\widehat{U}$ (see Equation 7.1). Because we calculate $\widehat{U}$ over all t-nodes, we need to account not only for agreement on tagging a t-node, but also for agreement on a t-node not being a part of a MWE, i.e. not tagged at all. This allows us to positively discriminate the cases where annotators agree that a t-node is not a part of a MWE from the cases where one annotator annotates a t-node and the other one does not, which is evidently worse.

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:

$$\widehat{U} = \frac{n_{A \cup B}}{N} + 0.051 \cdot \frac{N - n_{A \cup B}}{N} = 0.213. \tag{7.1}$$

The weight 0.051 used for scoring the t-nodes that were not annotated is explained below ($c = 4$). Because $\widehat{U}$ includes all the disagreements of the annotators, we believe that the real upper bound U lies somewhat below it and the agreement value 0.213 is not something that should (or could) be achieved. It is however based on the assumption that the data we have not yet seen have similar proportion of MWEs as the data we have used for the upper bound estimate.

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: $w_1 = 1$.

[1] The same calculations are done for our other pair: sta and vim.

$c = 2$ 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 look back at Equation 7.1 and see that $\widehat{U}$ is exactly the agreement which two ideal annotators reach.

It should be explained why we do not need to corrected upper bound when working with weighted measures like κ_w. There are weights for some types of disagreement in κ_w to distinguish "better" disagreement from "worse" one. But it is still a disagreement and annotators could agree completely. While in our task this class $c = 4$ represents agreement of its kind. The reason why we do not count it as an agreement is the biased resulting measure, if we do so.[2] The lesser they annotate the higher the agreement would be (with the extreme case of $\kappa = 1$ when they annotate nothing).

$c = 5$ 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 7.3.

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:

[2] We have also measured standard κ without weights. All partial disagreements were treated as full disagreements. In κ_1 we counted every non-annotated t-node as a disagreement, too; in κ_2 we think of non-annotation as a new category (with common agreement). And the difference is quite clear ($\kappa_1 = 0.04$ and $\kappa_2 = 0.68$) although κ is an agreement above chance and the expected agreement by chance was also different in κ_1 and κ_2.

$$\kappa_w^U = \frac{A_o - A_e}{\widehat{U} - A_e} = \frac{D_e - D_o}{\widehat{U} - 1 + D_e} \; . \tag{7.2}$$

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 7.3), 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

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

We introduced improved κ_w^U measure, which is 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.

7.4.2 Analysis of the disagreement

When we analyse disagreement and partial agreement we find that most cases have to do with SemLex entries rather than NEs. This is mostly due to the deficiencies

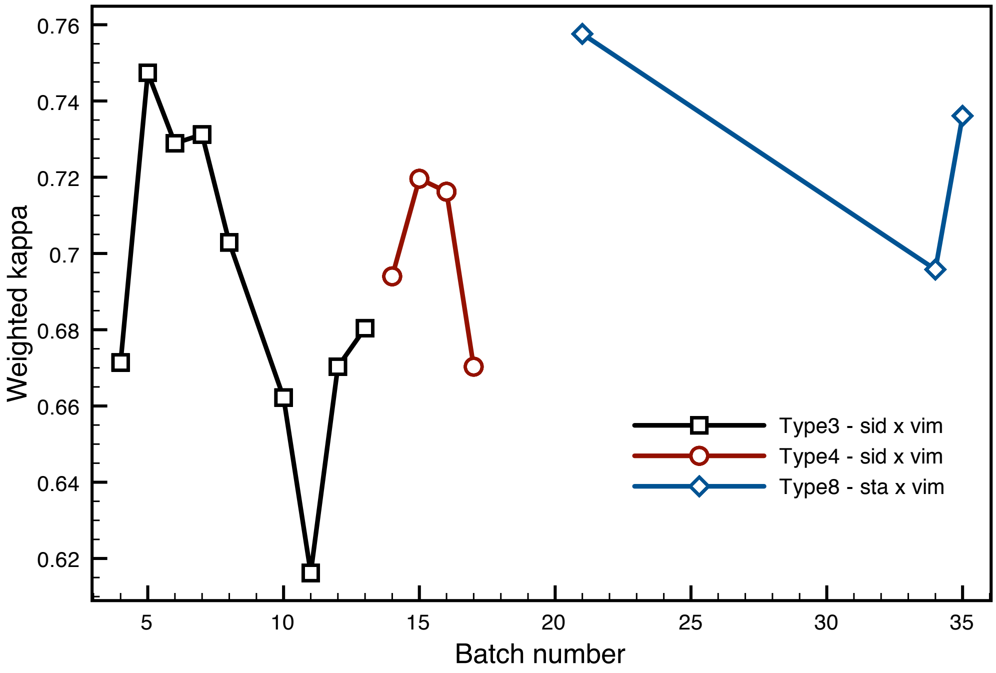


Figure 7.2: Weighted kappa per annotation type (colour line), a pair of annotators, and batches of annotated files (data points).

of the dictionary and its size (annotators could not explore each of almost 30,000 of SemLex entries). Our current methodology, which relies too much on searching the SemLex, is also to blame. This should, however, improve by employing pre-annotations (B) and (C).

One more reason for disagreement consists in the fact that there are cases for which non-trivial knowledge of the world is needed: "Jang Di Pertuan Agong Sultan Azlan Shah, the sultan of the state of Perak, [...] flew back to Perak." Is "Sultan Azlan Shah" still a part of the name or is it (or a part of it) a title?

The last important cause of disagreement is simple: both annotators identify *the same* part of text as MWE instances, but while searching the SemLex they choose different entry as the tags. This can be rectified by:

- Removing duplicate entries from SemLex (currently there are many almost identical entries originating from Eurovoc and Czech WordNet).
- Imploring improved pre-annotation B and C, as mentioned above.

7.5 Estimation of annotation intervals and speed

One of the reasons to implement detailed logs of all the annotations (see Section 5.2) was to allow detailed analysis of the time aspect of annotations. By "time aspect" we do not mean just checking, how much time it takes annotators to tag the data, even though this is an important information too. We wanted to make it possible to ask any number of questions: Is there correlation between annotation speed and frequency of work? Does the speed increase or decrease in long stretches of continuous work? In how long intervals do annotators tend to work? Is the number of tags per minute/hour more or less constant?

In Figure 7.3 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 analysed the speed of work during annotations to gain some basic insight into the time it actually takes annotators to create the amount of data they annotated. At this moment we must emphasise that we by no means imply some distrust within our project. Let the kind reader try to estimate how much time he/she spent this morning actually producing some work, be that writing a paper, writing a code, or anything else. It is by no means easy to estimate clean work time without some aid. So our aid were the log files.

We wrote a script tries to establish "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:

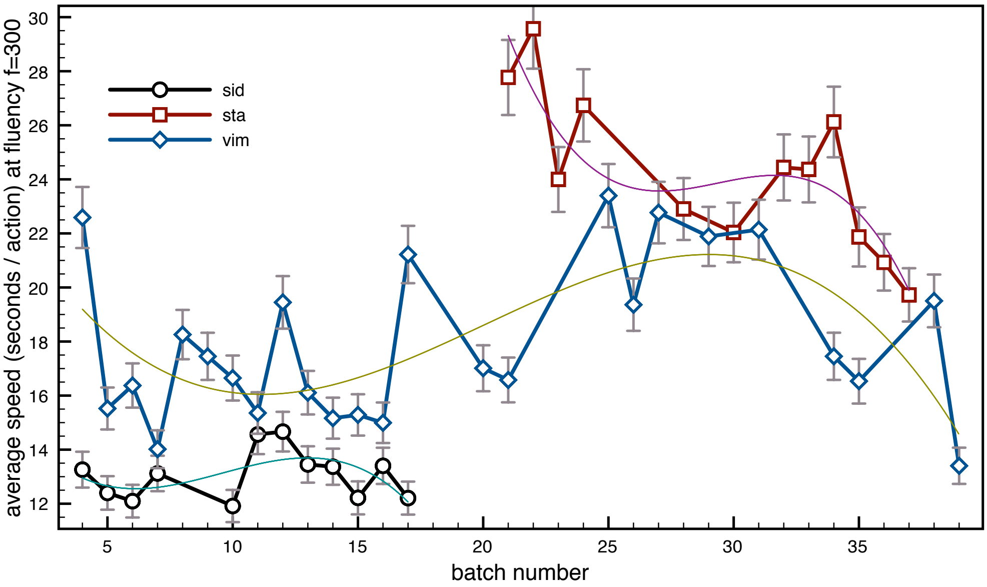


Figure 7.3: Average speed of each annotator for each batch that he/she annotated. Grey vertical bars show 5% error intervals.

$fluency$ 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 (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 7.3.

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

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

Chapter 8

Conclusions and Future work

8.1 Conclusions

Several years ago, we were thinking about the best way to begin working on improving the state of t-lemmas: to push them a bit forward, from the legacy of surface layer they still carry, towards deeper representation of lexical meaning. Our conclusion was, partly due to our previous experience with annotation using Czech WordNet as the annotation lexicon (Holub, 2003; Bejček et al., 2006), was to start by identifying 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 7.3 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 7.4.1 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, deriving it from Cohen's weighted kappa.

The resulting κ_w^U has actually 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 the hypotheses about tree structures of MWEs hold, provided the tectogrammatical layer is correctly annotated.[1] In this respect, our data, especially the places, where different t-structures were (on purpose!) 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/`.

We believe, however, that we still didn't manage to to properly process all the information that we have acquired during annotation and interesting work still remains to be done.

8.2 Future work

Considering the price of annotation, it is interesting, how much the annotation process itself stays out of focus of the researchers who create annotated data. Reading the papers and listening to presentations on NLP conferences and the various

[1] The only exception (there must always be one, after all) is in the coordinating conjunctions: since our MWE tree structures are built using the "effective parent" relation, the coordinating conjunctions are left apart, as they are not the effective parents of their (non-effective) children nodes.

annotation workshops, one cannot help but see this approach: The data is very interesting, so let us just create it somehow. Almost never is there any published information on the annotation process, factors that influence quality of data, or the price of the results.

Logs are a very good source of this valuable information. They can provide an insight into what is actually going on during annotation. Thorough statistical analysis of the logs can provide unbiased information that is hard to get any other way. Better analysis of logs together with s-files can perhaps improve estimation of the real cost of annotation. We have done this during annotation to some extent (see Section 7.5), but our method is not particularly sophisticated and we just more or less guessed the fluency constant for the annotation intervals.

As a small sample of what can be done with the data, we present some further information on the time aspect of annotation work. We decided to have a look at the distribution of times between two consecutive annotation actions[2]. We hoped that it can give us more information for properly handling the fluency of annotation intervals discussed in Section 7.5.

We leave it to real statisticians to examine the data more carefully, but our impression is that we used needlessly high value of fluency in our initial estimations described in Section 7.5. What is however quite striking in the plots in Figure 8.1 and Figure 8.2, is the similarity of the histograms. Even though they are not normalised to disregard the different amount of data annotated by each person, the distributions point to a remarkable similarity in behaviour of all three annotators. Compare for instance the distribution for the times up to five seconds. We do not have any explanation or even hypotheses for the uniform raise at 1 second[3], drop at 2, raise at three, etc. It seems to be an interesting point to examine, however.

But the analysis can go further and, to formulate the problem in economic terms, try to examine in general what factors influence the price of a (correct) tag. What is the relation of speed, length of work intervals, time of day, order of processing of the file, and other factors? We give only a very brief glimpse of one of the factors – speed of annotation, but in our opinion thorough statistical analysis of log files is an important source of information also for future annotation projects. It may be possible to maximise the efficiency of annotation by experimentally identifying the positive and negative factors. Some of the factors can be quite universal, such as worse concentration after n hours of work, but some may be quite individual (e.g. working in the morning, or on Saturdays…). If some positive and negative factors influencing annotation can indeed be identified, annotation guidelines, both for the project and for the individual annotator can be appropriately modified to maximise the positive and avoid the negative factors.

[2] A helpful suggestion of Zdeněk Žabokrtský.

[3] histograms actually indicate a lower bound of an interval. 2 seconds thus means $\langle 1, 2)$, etc.

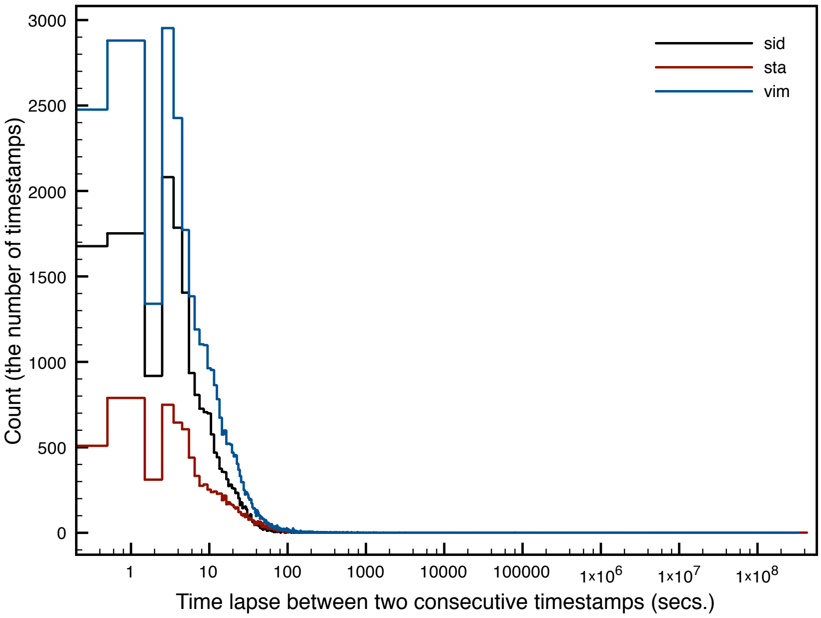


Figure 8.1: A histogram showing how many times (y) did an annotator place the next tag exactly x seconds after the previous one.

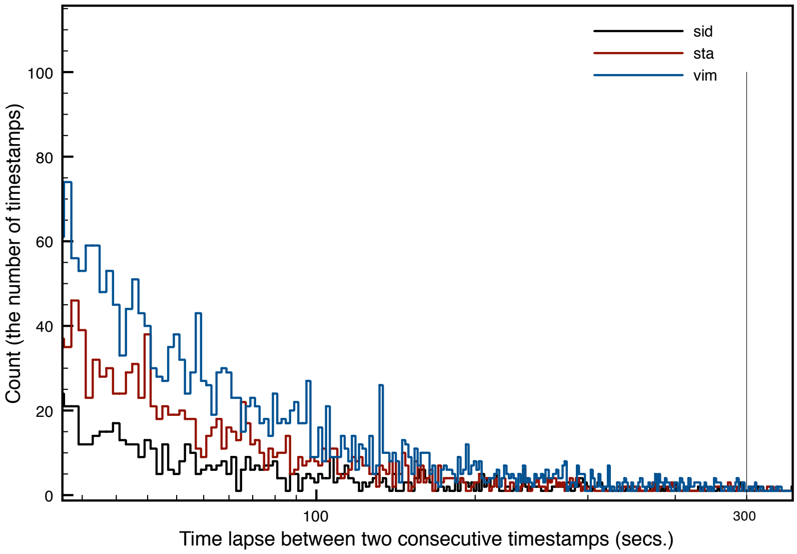


Figure 8.2: Detail of the histogram in Figure 8.1 in an interval where we have placed our *fluency* value for the preliminary experiment with clustering of work into annotation intervals ($f = 300$, see Section 7.5 and Figure 7.3).

It is of course possible that in the end no such factors can actually be estimated reliably, at least from our data. But currently we are in a position to seriously examine the possibility and to find out. That is more than has been done until now in any annotation project that we know of. From the little that we can see from our limited experiments, there is already some interesting data that requires interpretation.

Bibliography

Eurovoc, 2007. URL http://europa.eu/eurovoc/.

Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *extended version of the article submitted to Computational Linguistics*, 2007. URL http://cswww.essex.ac.uk/Research/nle/arrau/icagr.pdf.

Timothy Baldwin. Multiword expressions. CSSE, University of Melbourne, 2004. URL www.csse.unimelb.edu.au/~tim/pubs/altss2004.pdf.

Timothy Baldwin, Colin Bannard, Takaaki Tanaka, and Dominic Widdows. An empirical model of multiword expression decomposability. In *Proceedings of the ACL 2003 workshop on Multiword expressions*, pages 89–96, Morristown, NJ, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119282.1119294.

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

F. Čermák, V. Červená, M. Churavý, and J. Machač. *Slovník české frazeologie a idiomatiky*. Academia, 1994.

František Čermák. *Lexikon a sémantika*. Nakladatelství Lidové noviny, 2010.

Silvie Cinková, Josef Toman, Jan Hajič, Kristýna Čermáková, Václav Klimeš, Lucie Mladová, Jana Šindlerová, Kristýna Tomšů, and Zdeněk Žabokrtský. Tectogrammatical annotation of the wall street journal. *The Prague Bulletin of Mathematical Linguistics*, (92), 2009. ISSN 0032-6585.

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. *Disambiguation of Rich Inflection (Computational Morphology of Czech)*. Nakladatelství Karolinum, 2004. ISBN 80-246-0282-2.

Jan Hajič. *Insight into Slovak and Czech Corpus Linguistics*, chapter Complex Corpus Annotation: The Prague Dependency Treebank, pages 54–73. Veda Bratislava, Slovakia, 2005. ISBN 80-224-0880-8.

Jan Hajič, Jarmila Panevová, Zdeňka Urešová, Alevtina Bémová, Veronika Kolářová, and Petr Pajas. PDT-VALLEX. In Erhard Nivre, Joakim//Hinrichs, editor, *Proceedings of The Second Workshop on Treebanks and Linguistic Theories*, volume 9 of *Mathematical Modeling in Physics, Engineering and Cognitive Sciences*, pages 57–68, Vaxjo, Sweden, 2003. Vaxjo University Press. ISBN 91-7636-394-5.

Jan Hajič, Jarmila Panevová, Zdeňka Urešová, Alevtina Bémová, Veronika Kolářová, and Petr Pajas. PDT-VALLEX: Creating a large-coverage valency lexicon for treebank annotation. In Joakim Nivre and Erhard Hinrichs, editors, *Proceedings of The Second Workshop on Treebanks and Linguistic Theories*, volume 9 of *Mathematical Modeling in Physics, Engineering and Cognitive Sciences*, pages 57–68, Vaxjo, Sweden, 2003. Vaxjo University Press. ISBN 91-7636-394-5.

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.

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, and Magda Ševčíková-Razímová. Prague dependency treebank 2.0, 2006. URL `http://ufal.mff.cuni.cz/pdt2.0/`.

Jan Hajič, Silvie Cinková, Marie Mikulová, Petr Pajas, Jan Ptáček, Josef Toman, and Zdeňka Urešová. PDTSL: An annotated resource for speech reconstruction. In *Proceedings of the 2008 IEEE Workshop on Spoken Language Technology*, Goa, India, 2008. IEEE. ISBN 978-1-4244-3472-5.

Eva Hajičová, Barbara H. Partee, and Petr Sgall. *Topic-focus articulation, tripartite structures, and semantic content*, volume 71 of *Studies in Linguistics and Philosophy*. Kluwer, Dordrecht, 1998. ISBN 978-0-7923-5289-1.

Patrick Hanks. Norms and exploitations. a book manuscript, 2010.

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

Martin Holub. Sémanticko-lexikální model češtiny. printed note; working draft, 2 2003.

Martin Holub and Alena Böhmová. Use of dependency tree structures for the microcontext extraction. In *ACL-2000 Workshop on Recent Advances in Natural Language Processing and Information Retrieval*, pages 23–33, Hong Kong, China, October 2000. Association for Computational Linguistics. doi: 10.3115/1117755.1117759. URL http://www.aclweb.org/anthology/W00-1103.

Daniel Jurafsky and James H. Martin. *Speech And Language Processing*. Prentice Hall, 2. edition, 2008. URL http://www.cs.colorado.edu/~martin/slp2.html.

A. Kilgarriff. Senseval: An exercise in evaluating word sense disambiguation programs. In *Proc. LREC*, pages 581–588, Granada, 1998.

Igor Meľčuk. Lexical functions: A tool for the description of lexical relations in a lexicon. In Leo Wanner, editor, *Lexical Functions in Lexicography and Natural Language Processing*, volume 31 of *Studies in Language Companion Series*, pages 37–102. John Benjamins, 1996.

A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. The nombank project: An interim report. In A. Meyers, editor, *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

Rada Mihalcea. SEMCOR semantically tagged corpus, 1998. URL citeseer.ist.psu.edu/mihalcea98semcor.html.

Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Magda Razímová, Petr Sgall, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, and Zdeněk Žabokrtský. Annotation on the tectogrammatical level in the prague dependency treebank. annotation manual. Technical Report 30, ÚFAL MFF UK, Prague, Czech Rep., 2006.

Petr Pajas and Jan Štěpánek. 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., 2005.

Petr Pajas and Jan Štěpánek. PML toolkit, 2009. URL `http://ufal.mff.cuni.cz/jazz/PML/index_en.html`.

Karel Pala and Dana Hlaváčková. Derivational relations in czech wordnet. In *ACL '07: Proceedings of the Workshop on Balto-Slavonic Natural Language Processing*, pages 75–81, Morristown, NJ, USA, 2007. Association for Computational Linguistics.

Karel Pala and Pavel Smrž. Building czech wordnet. *Romanian Journal of Information Science*, 2004. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.100.7529`.

Martha Palmer, Dan Gildea, and Paul Kingsbury. The proposition bank: A corpus annotated with semantic roles. *Computational Linguistics Journal*, 31(1), 2005.

Pavel Pecina. *Lexical Association Measures: Collocation Extraction*, volume 4 of *Studies in Computational and Theoretical Linguistics*. Institute of Formal and Applied Linguistics, Prague, Czech Republic, 2009. ISBN 978-80-904175-5-7.

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/`.

Magda Ševčíková, Zdeněk Žabokrtský, and Oldřich Krůza. Zpracování pojmenovaných entit v českých textech. Technical Report TR-2007-36, ÚFAL MFF UK, Prague, Czech Republic, 2007.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. Academia/Reidel Publ. Comp., Praha/-Dordrecht, 1986.

Pavel Smrž. Quality control for wordnet development. In Petr Sojka, Karel Pala, Pavel Smrž, Christiane Fellbaum, and Piek Vossen, editors, *Proceedings of the Second International WordNet Conference—GWC 2004*, pages 206–212. Masaryk University Brno, Czech Republic, 2003.

David Yarowsky. One sense per collocation. In *Proceedings of the workshop on Human Language Technology*, HLT '93, pages 266–271, Morristown, NJ, USA, 1993. Association for Computational Linguistics. ISBN 1-55860-324-7. doi: http://dx.doi.org/10.3115/1075671.1075731. URL `http://dx.doi.org/10.3115/1075671.1075731`.