DISCOURSE AND COHERENCE

From the Sentence Structure to Relations in Text

Šárka Zikánová, Eva Hajičová, Barbora Hladká, Pavlína Jínová,
Jiří Mírovský, Anna Nedoluzhko, Lucie Poláková,
Kateřina Rysová, Magdaléna Rysová, Jan Václ

ÚFAL
ÚSTAV FORMÁLNÍ
A APLIKOVANÉ LINGVISTIKY

STUDIES IN COMPUTATIONAL AND THEORETICAL LINGUISTICS

Šárka Zikánová, Eva Hajičová, Barbora Hladká, Pavlína Jínová, Jiří Mírovský, Anna Nedoluzhko, Lucie Poláková, Kateřina Rysová, Magdaléna Rysová, Jan Václ

DISCOURSE AND COHERENCE
From the Sentence Structure to Relations in Text

Published by the Institute of Formal and Applied Linguistics
as the 14th publication in the series
Studies in Computational and Theoretical Linguistics.

The authors gratefully acknowledge the support of the grants GAP406/12/0658 (Coreference, discourse relations and information structure in a contrastive perspective), LM2010013 (LINDAT-CLARIN – Establishing and operating the Czech node of pan-European infrastructure for research), LH14011 (Multilingual Corpus Annotation as a Support for Language Technologies), P46 (PRVOUK – Programs of development of scientific areas at the Charles University in Prague: Informatics) and of the institutional funds of the Charles University in Prague.

Printed by Printo, spol. s r. o.

ISBN 978-80-904571-8-8

8

Searching in the PDT

Any large and richly annotated treebank would be of limited use if there was no way to effectively mine information from it, i.e. search for various phenomena that occur in the language and that have been annotated in the data. And if it is to be of value not only to computer scientists but also to (both theoretical as well as empirical) linguists, the search process needs to be simple and intuitive. The Prague Dependency Treebank is a very good example of a richly annotated treebank that poses a challenge for search tools. It contains annotations of several layers with non-trivial relations between some of them and with links to external resources (lexicons). For a manually annotated treebank, it is fairly large (50 thousand sentences annotated at all layers). The annotation is highly complex (the annotation guidelines for the tectogrammatical layer alone consist of more than twelve hundred pages). A tool that would allow for searching in and studying all annotated phenomena in the PDT has to be powerful in terms of the query language but simple to understand and use. Mírovský (2009) offers a study of what features a query language has to possess in order to be powerful enough for the PDT.

The PML-Tree Query (PML-TQ; Pajas and Štěpánek, 2009) is an advanced client–server system for searching in the Prague Dependency Treebank and other linguistically annotated treebanks encoded in the Prague Markup Language (PML; Hana and Štěpánek, 2012).[52] It offers a powerful query language with an intuitive, graphically oriented way of query creation.

Queries in the PML-TQ can be created both in a textual form and graphically. The basic (and simplified) idea of the system is such that a user draws a tree that should be included in a result tree as its subtree. The system processes the query and displays result trees one by one (if there are any), along with the context. The query language allows to define properties of tree nodes and relations among them (relations such as dependency, transitive dependency, left-right order, etc.) inside or between sentences and also across layers of annotation. Information from dictionaries (such as valency lexicons) can be easily incorporated. Negation and arbitrary logical constraints can be used in the queries. Results of the corpus search can be viewed one by one along

[52] Many existing treebanks have been transformed to the PML format and are searchable in the PML-TQ, including, for example, the Penn Treebank or the TIGER corpus. Also, in the project HamleDT (Zeman et al., 2014, see also https://ufal.mff.cuni.cz/hamledt), currently 30 dependency treebanks (or dependency conversions of other treebanks) have been harmonized into the same annotation style and are also searchable in the PML-TQ. For the list of treebanks available in the PML-TQ, see http://lindat.mff.cuni.cz/services/pmltq/.

with the context for a thorough inspection, or further processed with so-called output filters to produce statistical overviews. A detailed documentation can be found on the internet,[53] here we offer a simple introduction to the principal parts of the PML-TQ query language (Section 8.1) and show its usage on a set of illustrative discourse-related examples (Section 8.2). Section 8.3 gives technical details on how to download the PDT or how to access the public search server for the PDT.

8.1 Basics of the PML-TQ Language

8.1.1 Node selection

Values of attributes of a node can be set using several operators, mainly '=' for the equality relation, '~' for a regular expression, and 'in' for selection from a set of values. Each of these operators can be negated using the prefix '!'.

In Example 110, we are looking for sentences with an *Actor* atypically not expressed by a noun, i.e. for sentences like *It is difficult to live alone*, where the *Actor* (*to live*) is expressed by the infinitive verb form. The query consists of one tectogrammatical node and defines three of its attributes: The *functor* has to be an *Actor* (*ACT*), the semantic part of speech must not be a noun (i.e. it does not start with *n*), and the node does not have a substitute *t_lemma* (i.e. a *t_lemma* starting with #).[54]

(110a) The textual form of the query:

```
t-node
  [ functor = "ACT", gram/sempos !~ "^n", t_lemma !~ "^#" ];
```

(110b) The graphical form of the query:

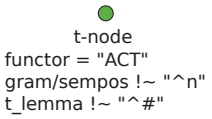


Figure 8.1 shows the tectogrammatical representation of one of the possible results. The matching node in the tree is highlighted in the same colour as the node in the query. In this case, the matching node is the node with *t_lemma být* [*to_be*], *functor ACT*, and semantic part of speech *v* (verb), i.e. the word *je* [*are*] in the subordinate clause, which is, as a whole, the *Actor* of the sentence 110c.

[53] http://ufal.mff.cuni.cz/pmltq/doc/pmltq_doc.html

[54] The substitute *t_lemma* is an artificial reconstruction of a lexical value of tectogrammatical nodes in the following cases: newly established nodes that are not copies of other nodes, personal and possessive pronouns, some types of punctuation marks and other symbols, and syntactic negation (see also Chapter 3, Section 3.5.3).

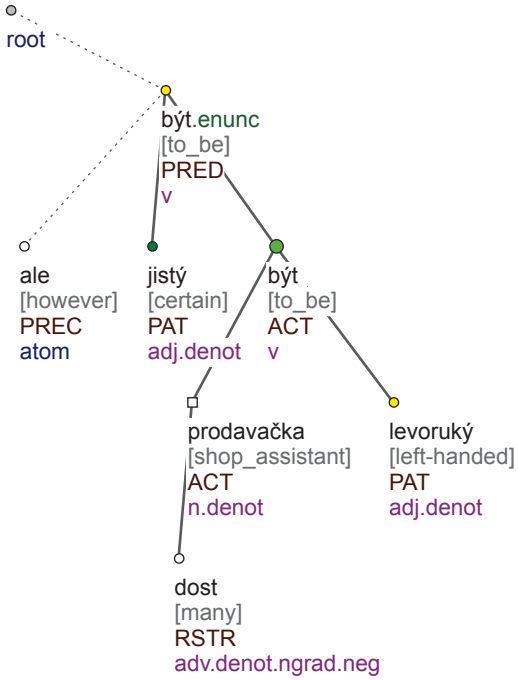


Figure 8.1: The tectogrammatical representation of the resulting sentence 110c for Example 110. The node matching the query is enlarged and highlighted in green (the same colour as the node in the query).

(110c) *Jisté ale je, že je dost levorukých* [*prodavaček*]. (PDT)
It is, however, certain that there are many left-handed [*shop assistants*].

8.1.2 Relations between nodes

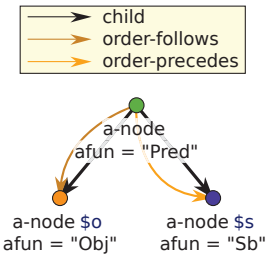
Various and multiple relations can be set between pairs of nodes in the query, including 'child', 'descendant', 'sibling', 'same-tree-as', 'same-document-as', but also 'order-follows', etc. The query in Example 111 searches at the analytical (surface syntax) layer of the PDT (see Chapter 6, Section 6.1) for sentences in the OVS (Object–Verb–Subject) order, i.e. for sentences such as *Ondru miluje Marie* [*Ondra is loved by Marie*, lit. *Ondra*$_{\text{Acc\_Obj}}$ *loves Marie*$_{\text{Nom\_Sb}}$]. The query searches for all *Predicates* that directly govern an *Object* and a *Subject*, and specifies that in the left-right order, the *Object* precedes the *Predicate* and the *Subject* is placed after the *Predicate*. These are language phenom-

ena represented at the analytical layer of the PDT, therefore we define analytical nodes (a-nodes) in the query.

(111a) The textual form of the query:

```
a-node
  [ afun = "Pred", order-follows $o, order-precedes $s,
    a-node $o :=
      [ afun = "Obj" ],
    a-node $s :=
      [ afun = "Sb" ] ];
```

(111b) The graphical form of the query:



In the textual version of the query, the first relation between two nodes can be (and usually is) defined by the recursive structure of the query, using square brackets, in this case (111a) with the implicit relation *child*. Additional relations between the same two nodes (e.g. the left-right order) need to be expressed using references to names of the nodes. In this query the *Object* is named *$o*, the *Subject* is named *$s*, and two additional relations are defined using references to these names – *order-follows $o* and *order-precedes $s*. In the graphical version of the query (111b), relations between nodes are expressed by coloured arrows. In the top part of the graphical version of the query, all different types of relations between nodes used in the query are listed, next to arrows in the respective colours.[55]

Figure 8.2 shows one of the results of the query. It is the analytical tree of the sentence 111c. Note that the *Object*, *Predicate* and *Subject* are in the required order.

(111c) *Rozepře prý zinscenoval tisk.* (PDT)
lit. *The_disputes*$_{\text{Acc\_Obj}}$ *allegedly staged the_press*$_{\text{Nom\_Sb}}$.
The disputes were allegedly staged by the press.

[55] The colours of these arrows do not correspond to colours of arrows representing non-dependency relations in the data, i.e. in the result trees, such as textual coreference or discourse relations.

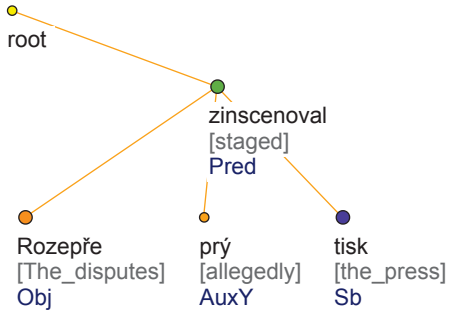


Figure 8.2: The analytical tree of the resulting sentence 111c for Example 111. Nodes matching the query are enlarged and highlighted in colours that match the nodes in the query.

8.1.3 Negative query

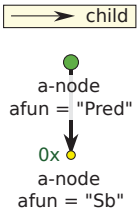
Negation on the level of relations between nodes is a very important part of the query language, as it allows to specify that "we do not wish something in the tree." With this kind of negation, it is possible to search, for example, for sentences without predicates, for predicates without subjects, or for contextually bound expressions without any referential link to the previous context. The PML-TQ uses so-called subqueries to specify how many times a part of the query tree should appear in the result tree (at a given place). "Zero times" then means that it should not be there at all.

The query in Example 112 searches at the analytical layer of the PDT for *Predicates* that do not directly govern a *Subject*, which is technically in the query expressed as a *Predicate* governing a *Subject* "zero times."

(112a) The textual form of the query:

```
a-node
  [ afun = "Pred",
    0x a-node
      [ afun = "Sb" ] ];
```

(112b) The graphical form of the query:



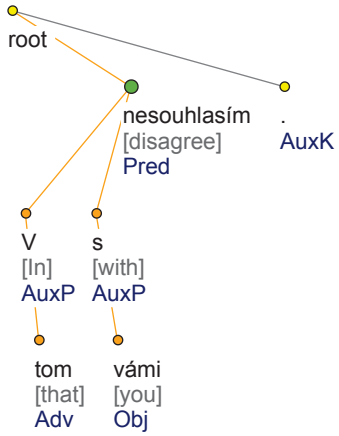


Figure 8.3: The analytical tree of the resulting sentence 112c for Example 112. The node matching the *Predicate* from the query is enlarged and highlighted in the matching colour, unlike, of course, the missing *Subject*.

(112c) *V tom s vámi nesouhlasím.* (PDT)
lit. *In that with you* [*I*] *disagree.*
I do not agree with you on that.

Figure 8.3 shows one of the results of the query. It is the analytical tree of the sentence 112c.[56] As Czech is a pro-drop language, no *Subject* is in this case expressed in the sentence (and neither in the analytical tree as a dependent node of the *Predicate*).[57]

8.1.4 Crossing the layers of annotation

Some queries need to combine information from various layers of annotation, for example to study surface syntax and morphology together or to study relations between the deep and surface representations of the sentence. Example 113 shows an inter-connection of the tectogrammatical layer and the analytical layer in a single query. We are looking at the tectogrammatical layer for a *Predicate* governing an *Actor*, which is not the *Subject* of the sentence in its representation at the analytical layer.

The query defines a t-node with the *functor PRED* and a depending t-node with the *functor ACT*. The connection from the *Actor* to its lexical counterpart at the analytical

[56] The English translations of the Czech word forms in the analytical trees are not a part of the treebank data. The translations have been added to the trees in the figures in this book for easier comprehensibility.

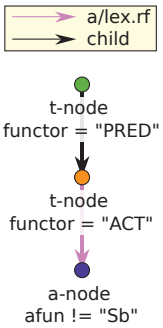
[57] The elided *Subject* of the sentence is reconstructed as an obligatory *Actor* at the tectogrammatical layer, see Chapter 6, Section 6.1.

layer is defined by using the attribute *a/lex.rf*, which is a link to the respective node at the analytical layer;[58] for this a-node, we require that it is not annotated as *Subject* (its analytical function *afun* is not *Sb*).

(113a) The textual form of the query:

```
t-node
  [ functor = "PRED",
    t-node [
      functor = "ACT",
      a/lex.rf a-node
        [ afun != "Sb" ] ] ];
```

(113b) The graphical form of the query:



In one of the results (see the sentence 113c and its tectogrammatical representation in Figure 8.4), the t-nodes from the query match the node representing the passive verb form *je vázána* [*is bound*], together with the dependent node representing the word *dohodami* [*by agreements*]. At the analytical layer, *dohodami* [*by agreements*] is an *Object*.

(113c) *K zákazu je ČR vázána mezinárodními dohodami.* (PDT)
lit. *To the_ban is ČR bound* [*by*] *international agreements.*
The Czech Republic is bound to [*implement*] *the ban by international agreements.*

8.2 Discourse Coherence Phenomena and the PML-TQ

8.2.1 Non-dependency relations

Textual coreference, bridging anaphora and discourse relations (among others) are represented in the data as references from one node (*start node* – the node where the

[58] There is at most one lexical analytical counterpart for each t-node, represented by its identifier in the attribute *a/lex.rf*; for auxiliary analytical counterparts (a-nodes representing prepositions, modal verbs etc.), there is a list of their identifiers in the attribute *a/aux.rf*.

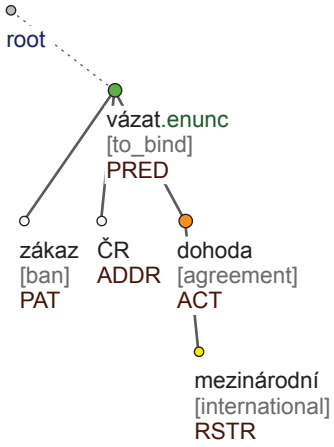


Figure 8.4: The tectogrammatical representation of the resulting sentence 113c for Example 113

relation starts) to another node (*target node* – the node where the relation ends).[59] In the graphical representation of the trees, these relations are depicted as curved arrows connecting the respective two nodes. As several relations of each type may start at a single node and as these relations carry additional information (e.g. the discourse type, scope of the arguments), they are represented in the query language of the PML-TQ as special *member* nodes.

Example 114 shows how to search for a discourse relation of a given type.[60] The query defines two t-nodes connected with a member node that stands for a discourse relation between arguments represented by the two nodes. The required type of the discourse relation can be specified at the member node – in this case it is set to *reason*. The query also specifies that the start and target nodes of the relation are not from the same tree, i.e. it looks for an inter-sentential discourse relation of the type *reason–result*.

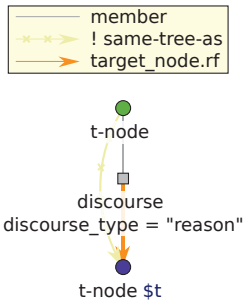
[59] The same technique is used for other relations, like the secondary relation in the verbal complement, or – as shown in Example 113 – the connection between layers of annotation.

[60] Searching for the textual coreference or the bridging anaphora would be very similar, using the respective type of the member node.

(114a) The textual form of the query:

```
t-node
  [ !same-tree-as $t,
    member discourse
      [ discourse_type = "reason",
        target_node.rf t-node $t := [ ] ] ];
```

(114b) The graphical form of the query:



The following two sentences represent one of the results of the query.

(114c) *Neprošel s ní celnicí.* ***Tak***$_{\textbf{reason–result}}$ *si ji pověsil ve své hospodě na stěnu.*[61] (PDT)
lit. *He_did_not_get with it through_customs.* ***So***$_{\textbf{reason–result}}$ REFL *it hung in his pub on the_wall.*
He could not take it through customs. ***So***$_{\textbf{reason–result}}$ *he has hung it on the wall in his pub.*[61]

Figure 8.5 captures the tectogrammatical annotation of these two sentences, along with the discourse relation represented by the thick orange arrow connecting roots of the two respective propositions. Additional relevant information is displayed also in orange at the start node (type of the relation, the connective and the range of the arguments).

8.2.2 Topic-focus articulation and anaphora

The following Example 115 combines some of the techniques described so far to search for a phenomenon studied later in detail in Chapter 13. Specifically, we are interested in non-contrastive contextually bound nodes from which there is no anaphoric reference to the previous context.[62] The query defines a t-node with *tfa* value *t*, from which there is no link of grammatical coreference, no link of textual coreference, and

[61] To give the reader a bit of context, the story is about climbers discussing the perfect prosthesis.
[62] nor any cataphoric or exophoric reference

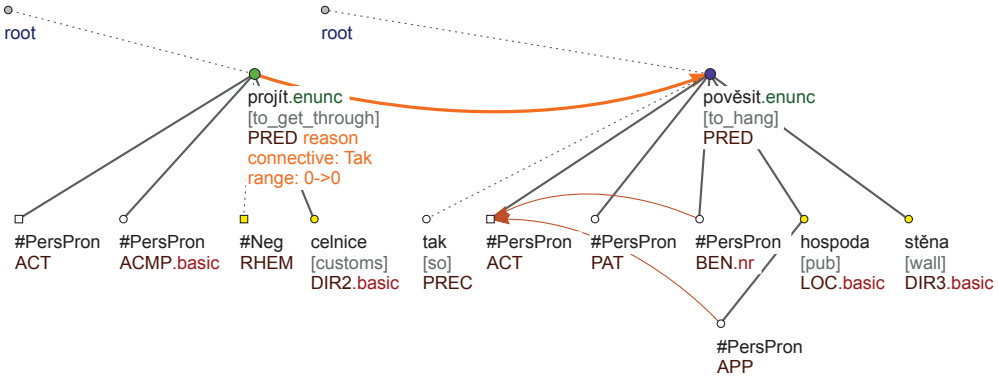


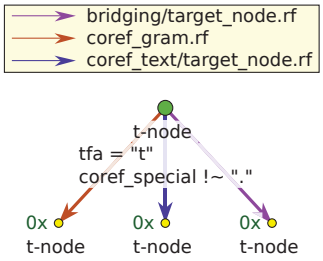
Figure 8.5: The tectogrammatical representation of two resulting sentences 114c for Example 114

no link of bridging anaphora. There may also be no link to an unspecified previous segment and no exophoric link either (both would be captured in the attribute *coref_special* as values *segm* and *exoph*, respectively).

(115a) The textual form of the query:

```
t-node
  [ tfa = "t",
    coref_special !~ ".",
    0x coref_gram.rf t-node [ ],
    0x coref_text/target_node.rf t-node [ ],
    0x bridging/target_node.rf t-node [ ] ];
```

(115b) The graphical form of the query:



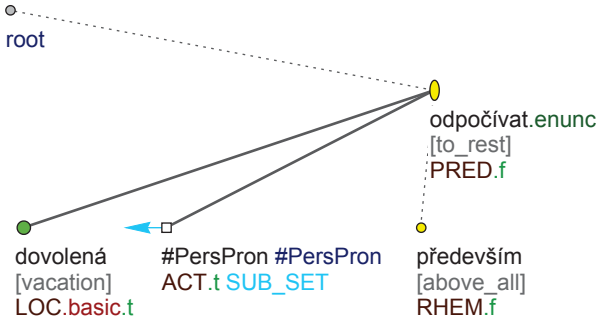


Figure 8.6: The tectogrammatical representation of the resulting sentence 115c for Example 115. Values of the attribute *tfa* are displayed in green next to the *functor*.

(115c) ***Na dovolené*** *chceme především odpočívat.*[63] (PDT)
On vacation*, we want above all to rest.*[63]

Figure 8.6 shows the tectogrammatical representation of the resulting sentence 115c. It is the second sentence of a document and a subheading[64] of the article with an immediately preceding heading 115d. For the reader, the word *dovolená* [*vacation*] is somehow connected to the previous sentence and can be considered contextually bound; however this type of relation is not in any way captured in the PDT annotation.

(115d) *Pojedete do zahraničí s cestovkou?* (PDT)
Will you go abroad with a travel agency?

8.2.3 Output filters

Results of queries can be further processed using *output filters*. Thanks to an output filter, a result of a query does not consist of individual occurrences of the query in the data but instead of a summary of all its occurrences in the searched data, specified by the output filter and presented as a table.[65]

In Example 116, an output filter is added to a simple search. The query defines a single t-node, which is required to be an *Actor* but not a semantic noun (its grammateme *gram/sempos* does not start with *n*) and it does not have a substitute *t_lemma*

[63] a sentence of high significance to the authors of the present book, as the volume was finished during the summer months of 2015
[64] as indicated by the value *heading* in the attribute *discourse_special* and graphically expressed by the oval shape of the node *odpočívat* [*to_rest*]
[65] The result of the output filter can be saved to a textual file in the CSV (comma-separated values) format.

(it does not start with # like e.g. *#PersPron*). The output filter is defined on the last line of the textual form of the query, after the sign '>>'. It states that for each value of the attribute *gram/sempos* found at all nodes matching the query node *$t*, the value of the grammateme (*$1* refers to *$t.gram/sempos*) along with its total count should be listed, and the results should be sorted by the count (referred to by *$2*) in the descending order, i.e. from the most frequent semantic part of speech to the least frequent one.

(116a) The textual form of the query:

```
t-node $t :=
  [ functor = "ACT", gram/sempos !~ "^n", t_lemma !~ "^#" ];

>> for $t.gram/sempos give $1, count() sort by $2 desc
```

(116b) The graphical form of the query:

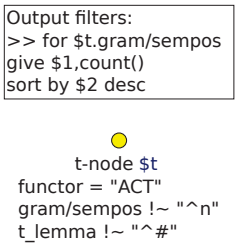


Table 8.1 shows the result produced by the query 116 with the output filter.[66] In the left column of the table, the semantic part of speech is listed, in the right column, numbers of occurrences of the respective semantic parts of speech are presented.

8.2.4 Output filters in discourse

Example 117 is similar to Example 114, except that the condition on discourse type has been removed and an output filter has been added. The query searches for all inter-sentential discourse relations (of any discourse type) and the output filter (very similar to the output filter from the previous example) summarizes the results in a distribution of discourse types in the relations.

[66] Numbers in tables in this chapter correspond to a search in 9/10 of the Prague Dependency Treebank, available at the public search server. The remaining 1/10 of the data serve as test development data and therefore are not accessible this way.

gram/sempos	Number of occurrences
v	3,028
adj.quant.grad	75
adv.denot.grad.neg	31
adj.denot	28
adv.pron.def	24
adj.quant.def	15
adv.denot.ngrad.nneg	4
adv.denot.ngrad.neg	2
adj.pron.def.demon	2
adj.quant.indef	1

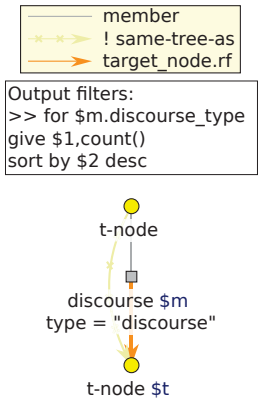
Table 8.1: The resulting table for Example 116. The value *v* stands for a semantic verb, values starting with *adj* are semantic adjectives, and values starting with *adv* semantic adverbs. Semantic adjectives and adverbs are further subcategorized, for example *adj.quant.grad* means a gradable quantificational semantic adjective (adjectives such as *mnoho* [*many*]). Headings of the columns are not a part of the query result.

(117a) The textual form of the query:

```
t-node
  [ !same-tree-as $t,
    member discourse $m :=
      [ type = "discourse",
        target_node.rf t-node $t := [ ] ] ];

>> for $m.discourse_type give $1, count() sort by $2 desc
```

(117b) The graphical form of the query:



Discourse type	Number of occurrences
opp	1,601
conj	1,255
reason	902
confr	272
conc	236
preced	215
grad	184
restr	149
explicat	121
corr	110
...	

Table 8.2: First 10 rows in the resulting table for the Example 117 (headings of the columns are not a part of the query result).

Table 8.2 shows the result produced by the query 117 with the output filter.

A more advanced output filter is used in Example 118, which shows how output filters (or, in other words, lines of an output filter) can be put one after another – the second line of an output filter is applied to the output of the first one, etc. The query 118 searches for all discourse relations in the data and the output filter summarizes the results in a distribution of all, intra- and inter-sentential usages of connectives in the relations (regardless of the discourse types they represent), both in total counts and percentages.

The first line of the output filter produces a table[67] with a row for each discourse relation matching the query, consisting of three values (i.e. the output table has three columns) – a lowercased connective along with the information whether the given relation is intra-sentential (the condition on tree numbers of the start and target nodes produces *1* in the second column) or inter-sentential (*1* in the third column). The second line of the output filter (applied to the output of the previous line) adds up the intra-sentential and inter-sentential occurrences of relations for the various connectives (*$1*, *$2* and *$3* refer to the respective columns in the result of the previous line of the output filter), and for the purpose of counting percentages of these numbers adds the fourth column – a total number of occurrences of the given connective in the relations (intra- and inter-sentential ones together). The third line (applied to the output of the second line) counts the percentages and formats the output (by reorganizing the order of columns and by adding parentheses and percentage marks).

[67] a temporary table, further processed by the subsequent lines of the output filter

(118a) The textual form of the query:

```
t-node $s :=
  [ member discourse $m :=
    [ type = "discourse", target_node.rf t-node $t := [ ] ] ];
>> give lower($m.connective), if(tree_no($s) = tree_no($t),1,0), if(tree_no($s)
= tree_no($t),0,1)
>> for $1 give distinct $1, sum($2), sum($3), sum($2)+sum($3)
>> give $1,$4,$2,"(" & $2 * 100 div $4 & "%)",$3,"(" & 100 - ($2 * 100 div $4) &
"%)" sort by $2 desc
```

(118b) The graphical form of the query:

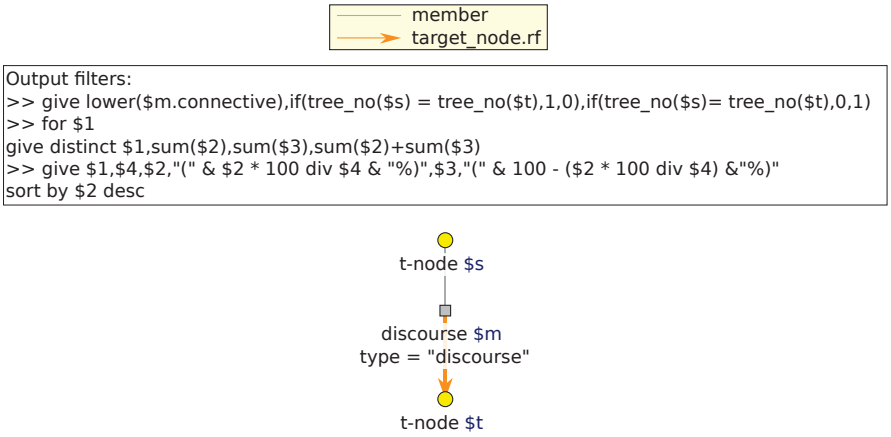


Table 8.3 shows the result produced by the query 118 with the output filter.

8.3 Hands on the Data

The last section of this chapter is dedicated to a highly technical matter – how to get the data – and can be safely skipped if the reader is not interested in either downloading the PDT or searching in it using the PML-TQ query language described above.

8.3.1 Data to download

The PDT 3.0 is freely available from the public repository of the Lindat/Clarin project[68] under the Creative Commons Licence.[69] The data are stored in the Prague Markup Language format (PML; http://ufal.mff.cuni.cz/jazz/PML/), which is an XML-based format designed to capture complex annotations of language data, particularly tree-

[68] http://hdl.handle.net/11858/00-097C-0000-0023-1AAF-3

[69] Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported Licence (CC BY-NC-SA 3.0)

Connective	Total	Intra-sentential	(%)	Inter-sentential	(%)
a [*and*]	5,128	4,815	(93%)	313	(7%)
však [*however*]	1,356	236	(17%)	1,120	(83%)
ale [*but*]	1,134	758	(66%)	376	(34%)
když [*when*]	478	478	(100%)	0	(0%)
protože [*because*]	469	463	(98%)	6	(2%)
totiž [*actually, in fact*]	405	20	(4%)	385	(96%)
:	353	310	(87%)	43	(13%)
pokud [*if*]	342	342	(100%)	0	(0%)
proto [*therefore*]	339	32	(9%)	307	(91%)
aby [*to*]	276	275	(99%)	1	(1%)
tedy [*therefore*]	269	30	(11%)	239	(89%)
pak [*then*]	257	66	(25%)	191	(75%)
ovšem [*however*]	257	57	(22%)	200	(78%)
li [*if*]	227	227	(100%)	0	(0%)
také [*also*]	208	7	(3%)	201	(97%)
neboť [*because*]	196	196	(100%)	0	(0%)
–	194	193	(99%)	1	(1%)
zatímco [*while*]	175	174	(99%)	1	(1%)
nebo [*or*]	169	150	(88%)	19	(12%)
navíc [*moreover*]	169	24	(14%)	145	(86%)
...					

Table 8.3: First 20 rows in the resulting table for Example 118 (headings of the columns and the translations are not a part of the query result).

banks. Although the format is text-based, it is difficult to read in the raw form, more so at the higher layers of annotation. The tree editor TrEd (see below) should be used for more convenient viewing.

Tree editor TrEd

Tree editor TrEd (Pajas and Štěpánek, 2008) is a primary tool for browsing (and editing) the PDT 3.0 data. It can be freely downloaded (for various platforms, including MS Windows and Linux) from its home web page[70] under the GPL – The General Public Licence. After the installation of the editor itself, an extension for the PDT 3.0

[70] http://ufal.mff.cuni.cz/tred/

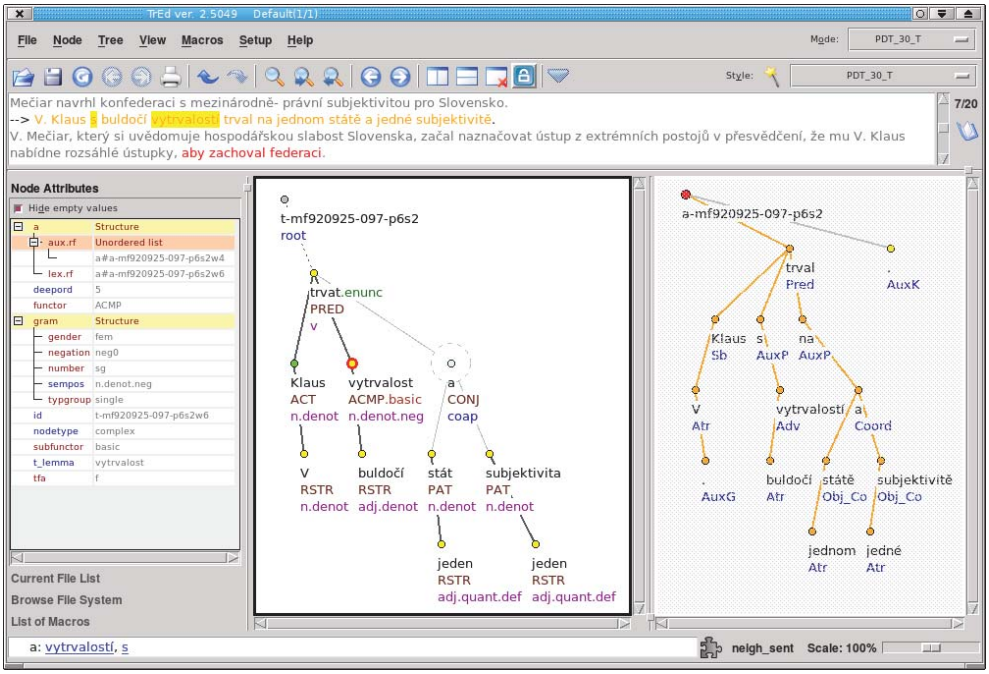


Figure 8.7: The tectogrammatical and analytical representations of the sentence *V. Klaus s buldočí vytrvalostí trval na jednom státě a jedné subjektivitě* [*V. Klaus with bulldog persistence insisted on a single state and a single* [*legal*] *subjectivity*], displayed in TrEd.

support needs to be installed as well, which can be done from the TrEd menu using Setup → Manage Extensions → Get New Extensions.[71]

Afterwards, any document from the PDT can be opened in TrEd. Annotation of each document is stored in four files corresponding to the word layer (file with the suffix *.w*), the morphological layer (file with the suffix *.m*), the analytical layer (file with the suffix *.a*), and the tectogrammatical layer (file with the suffix *.t*).[72] Any of these files except for the files of the word layer can be opened in TrEd; for displaying the tectogrammatical annotation of a given document, the respective file with the suffix *.t* needs to be opened, the file with the suffix *.a* for the analytical annotation, and the file with the suffix *.m* for the morphological annotation.[73] Figure 8.7 shows

[71] A more detailed description of the installation can be found in the documentation for the PDT 3.0 at http://ufal.mff.cuni.cz/pdt3.0/data.

[72] All files are compressed by gzip, which means that their suffixes are in fact *.w.gz*, *.m.gz*, *.a.gz* and *.t.gz*.

[73] TrEd is a tree editor. Therefore, it cannot be directly used to open files of the word layer. However, thanks to the interlinking of the layers, the information at the w-layer is accessible from the higher layers. Files

the graphical interface of TrEd, which consists of several sections for displaying the data, namely the textual area on top (it displays the sentence in its context) and – in this case – two areas for the tectogrammatical and analytical representations of the given sentence (in the middle and on the right). Values of all non-empty attributes for a selected node are displayed in the left panel.

8.3.2 Data for searching

The PDT 3.0 data can be searched on a public search server, i.e. without a prior download, using the PML-TQ – the query language described in the previous sections.

There are two ways of accessing the search server for the PDT. The first method uses the tree editor TrEd along with a PML-TQ extension.[74] The second method accesses the server using a web browser; the server for PDT 3.0 data is publicly available at the LINDAT/CLARIN portal.[75] The web-based access has several limitations, namely less variability in displaying the results and the necessity to create the query in the textual form. TrEd, on the other hand, needs to be installed first (along with the PML-TQ extension), but it does offer a better user interface, the query can be created graphically and the graphical representation of the results can be adapted to the user's needs.

We have introduced two possible ways how to get one's hands on the data of the Prague Dependency Treebank – downloading the corpus or searching in it on-line. As such, the data are open to experiments and easily accessible for studying many language phenomena. It is very important that various levels of annotation/language description are annotated separately but can be used together, even in a single search query. It opens the possibility for researchers to study – if we stay on the topic of this book – the interplay among morphology, the syntactic structure of the sentence, discourse relations, anaphora, and the topic–focus articulation. Several such case studies are presented in the subsequent part of the book.

---

of the morphological layer can be opened in TrEd, because, technically, each sentence is at this layer represented as a sequence of nodes corresponding to the words of the sentence, with a single technical root as their common parent. This solution would not be practical for the w-layer, as the data at the w-layer are not segmented into individual sentences.

[74] See the on-line documentation of the PML-Tree Query for instructions on how to install the extension: http://ufal.mff.cuni.cz/pmltq/.

[75] https://lindat.mff.cuni.cz/services/pmltq/