Towards Semantic Annotation Supported by Dependency Linguistics and ILP

Jan Dědek

Department of Software Engineering, Faculty of Mathematics and Physics,
Charles University in Prague, Czech Republic

ISWC 2010, Nov 9, 2010, Shanghai, China
Doctoral Consortium

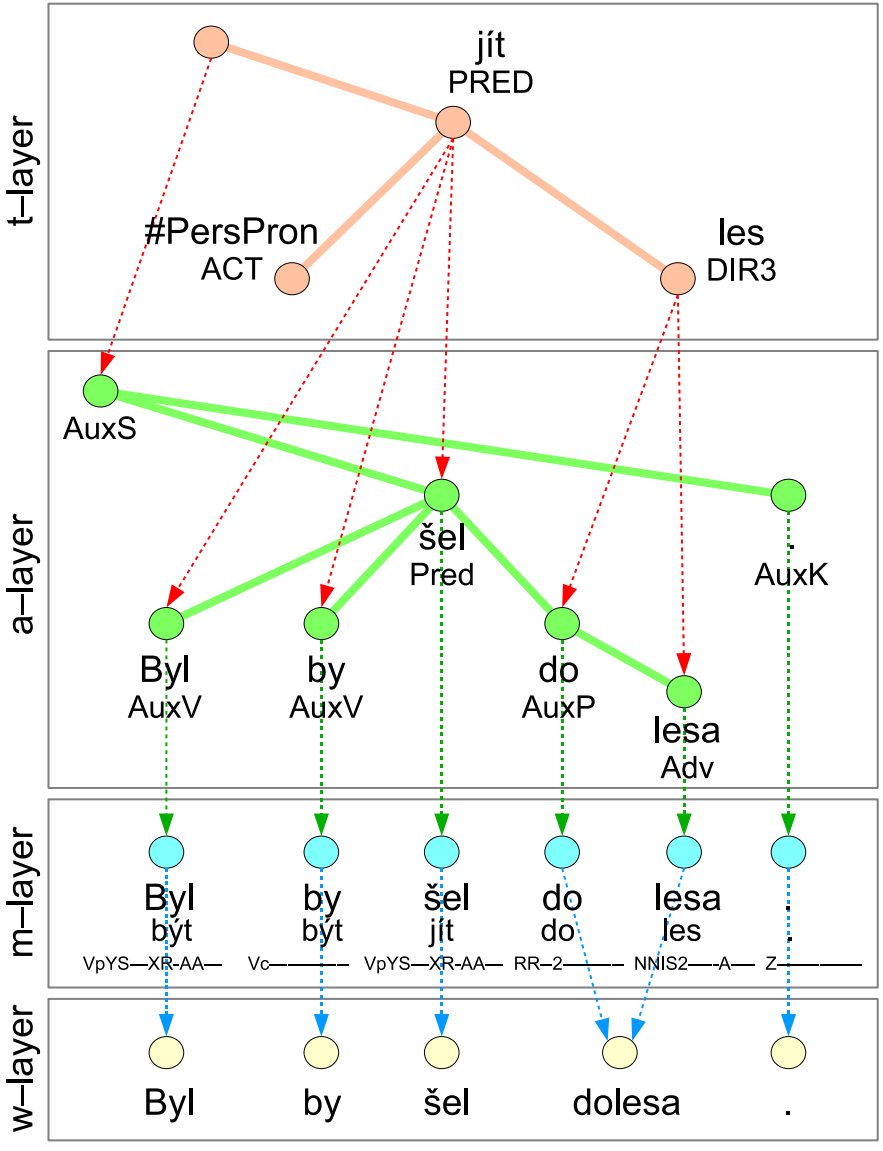
Outline

Motivation

1. The goal of the Semantic Web evolution
   - Bring structure to unstructured resources
2. Complex linguistic tools
   - The challenge of machine understanding of text

- Using linguistics and Information Extraction for the Semantic Web
  - Semantic annotation

Layers of linguistic annotation in PDT



- Tectogrammatical layer “is supposed to represent the semantic structure of a sentence”
- Analytical layer (syntax)
- Morphological layer
- PDT 2.0 on-line:

`http://ufal.mff.cuni.cz/pdt2.0/`

Sentence:

Byl by šel dolesa.
He-was would went toforest.

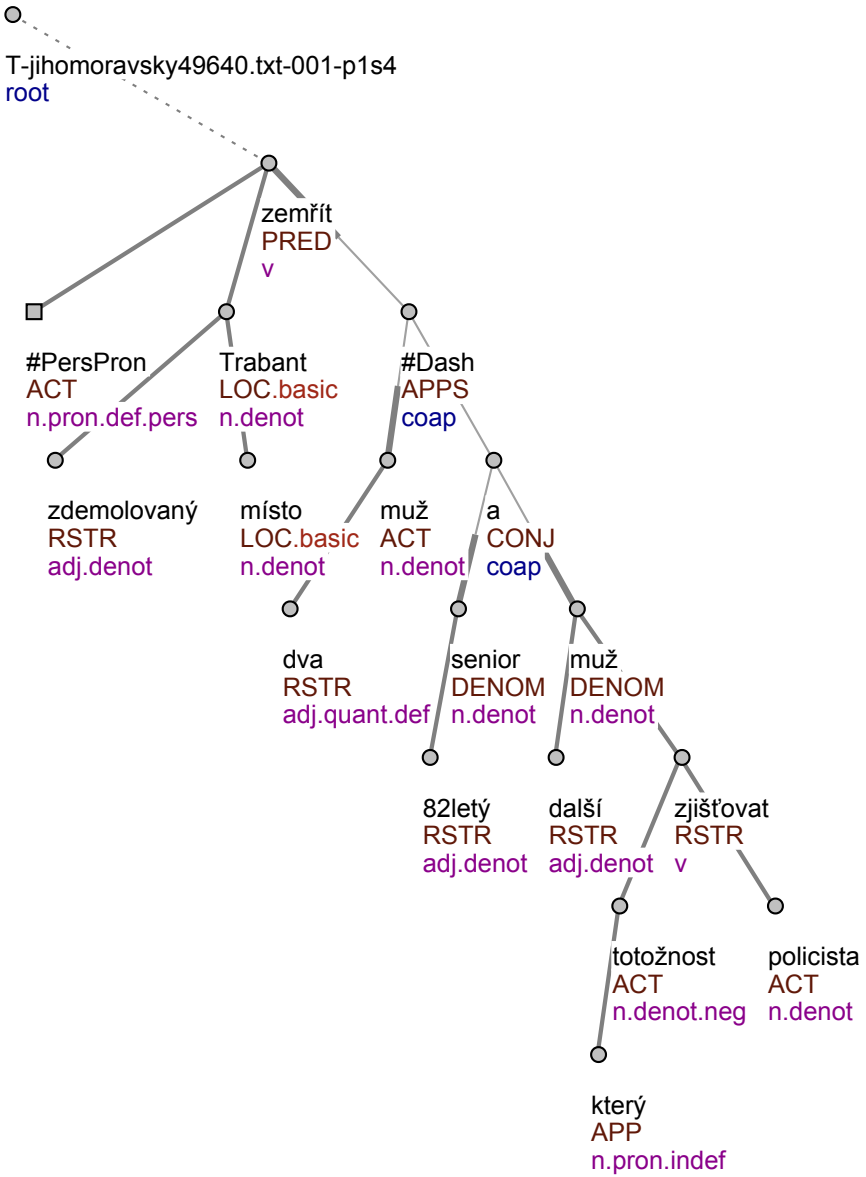
Tools for machine linguistic annotation

1. Segmentation and tokenization
2. Morphological analysis
3. Morphological tagging
4. McDonnald's Maximum Spanning Tree parser
   – Czech adaptation
5. Analytical function assignment
6. Tectogrammatical analysis
   – Developed by Václav Klimeš

- Available within the TectoMT[1] project

[1] http://ufal.mff.cuni.cz/tectomt/

Example of tectogrammatical tree



- Lemmas
- Functors
- Semantic parts of speech

Sentence:

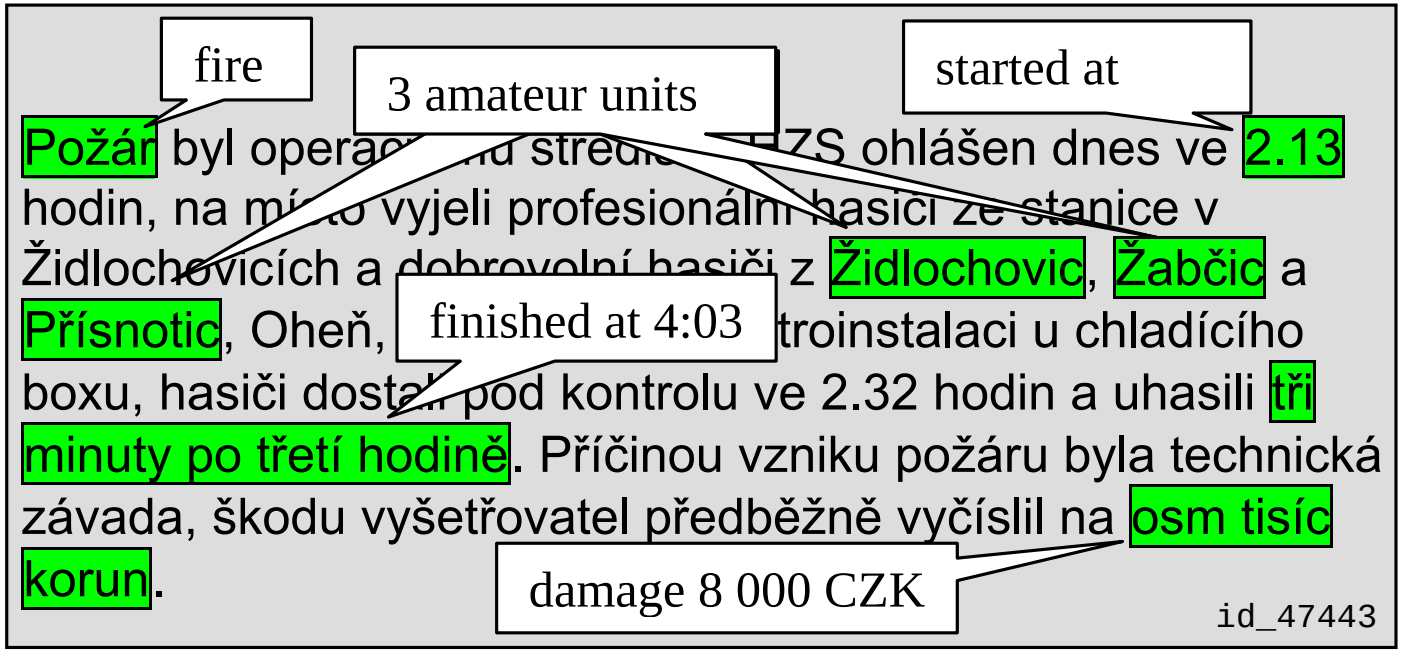
Ve zdemolovaném trabantu na místě zemřeli dva muži – 82letý senior a další muž, jehož totožnost zjišťují policisté.

Two men died on the spot in demolished trabant – . . .

Introduction to the Presented Work

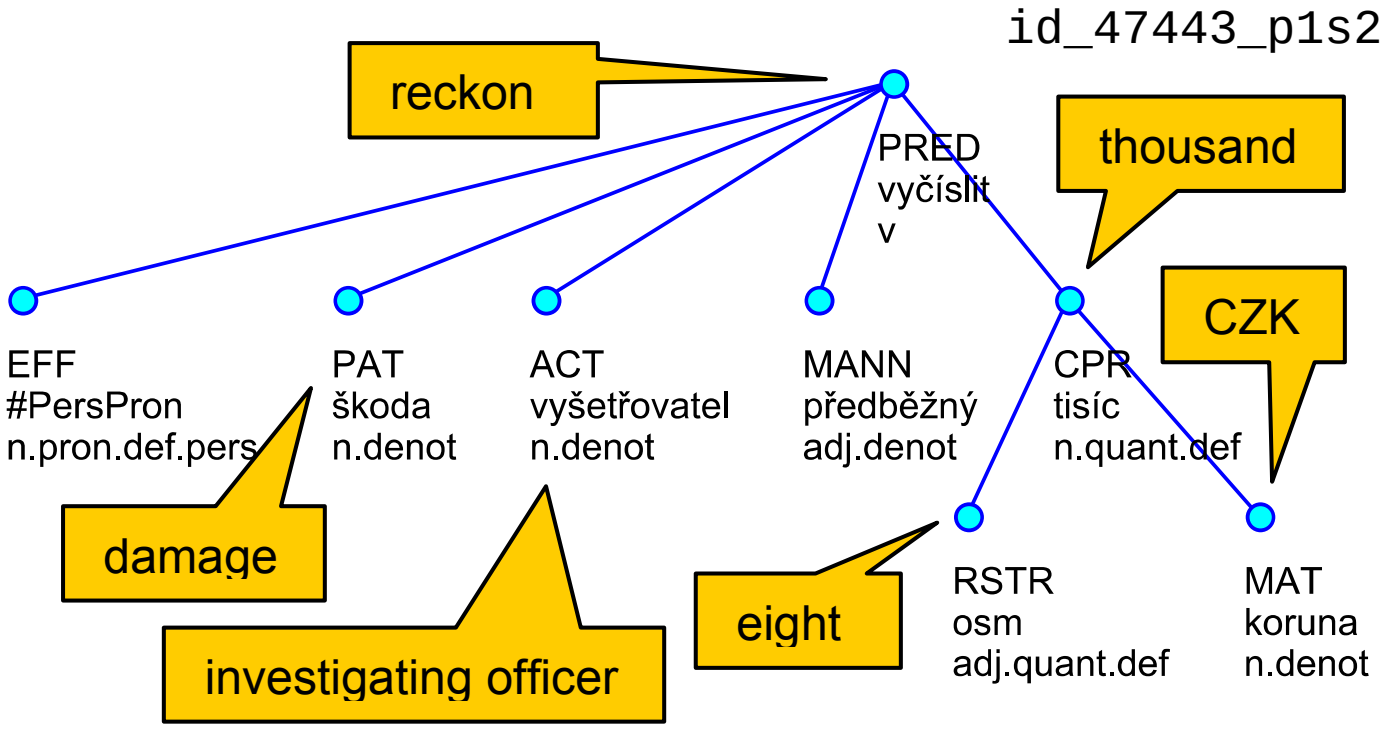
- Extraction of semantic information from texts.
  - In Czech language.
  - Coming from web pages.
- Exploiting of linguistic tools.
  - **Prague Dependency Treebank** project.
  - **TectoMT** project (ÚFAL MFF UK).
  - **GATE** project (The University of Sheffield).
  - Experiments with the **Czech WordNet**.
- Rule based extraction method.
  - Extraction rules $\approx$ linguistic tree queries
  - ILP learning of extraction rules

Example of processed text – a fire accident



- Information to be extracted is decorated.
- See the last sentence on the next slide.

Example of a linguistic tree

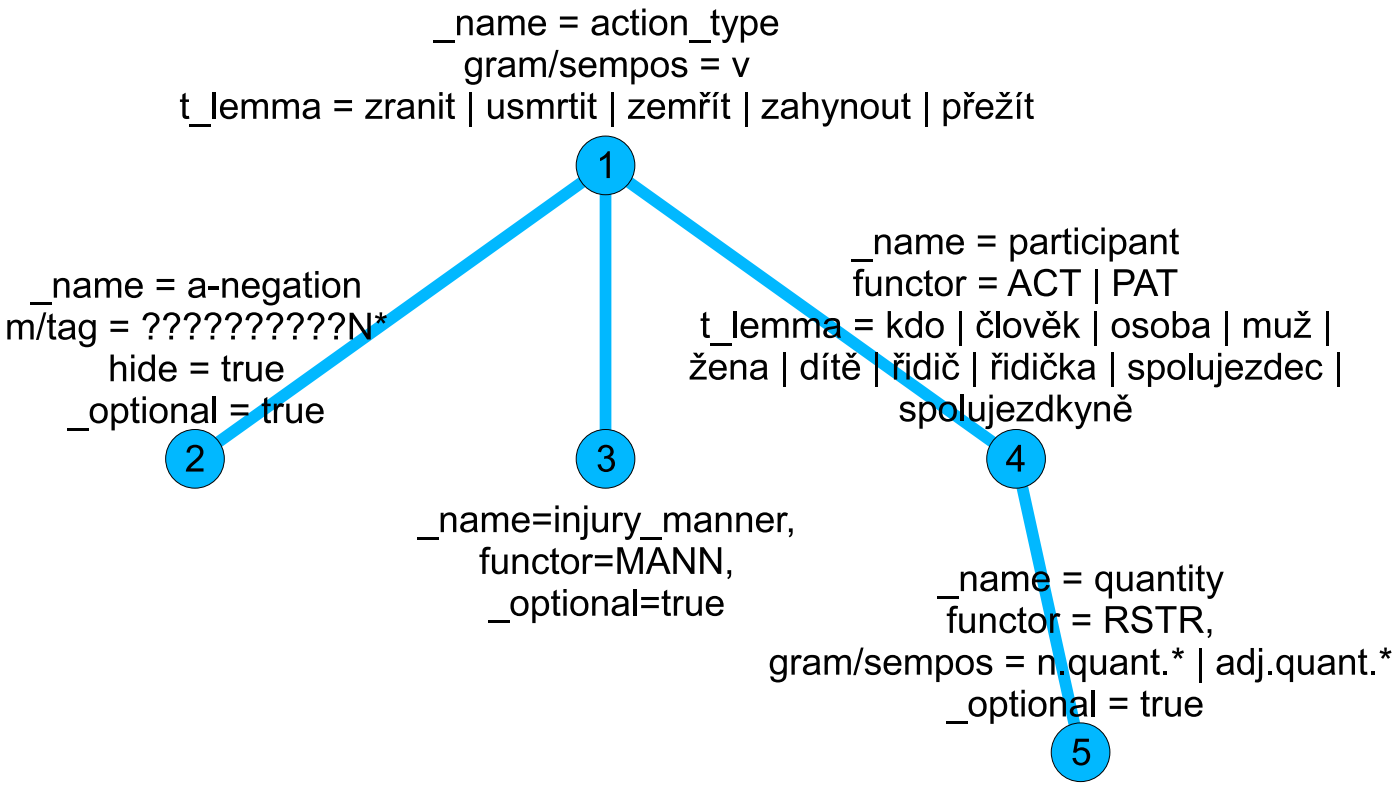


…, škodu vyšetřovatel předběžně vyčíslil na osm tisíc korun.

…, investigating officer preliminarily reckoned the damage to be 8 000 CZK.

- Our IE method uses tree queries (tree patterns)

Extraction rules – Netgraph queries



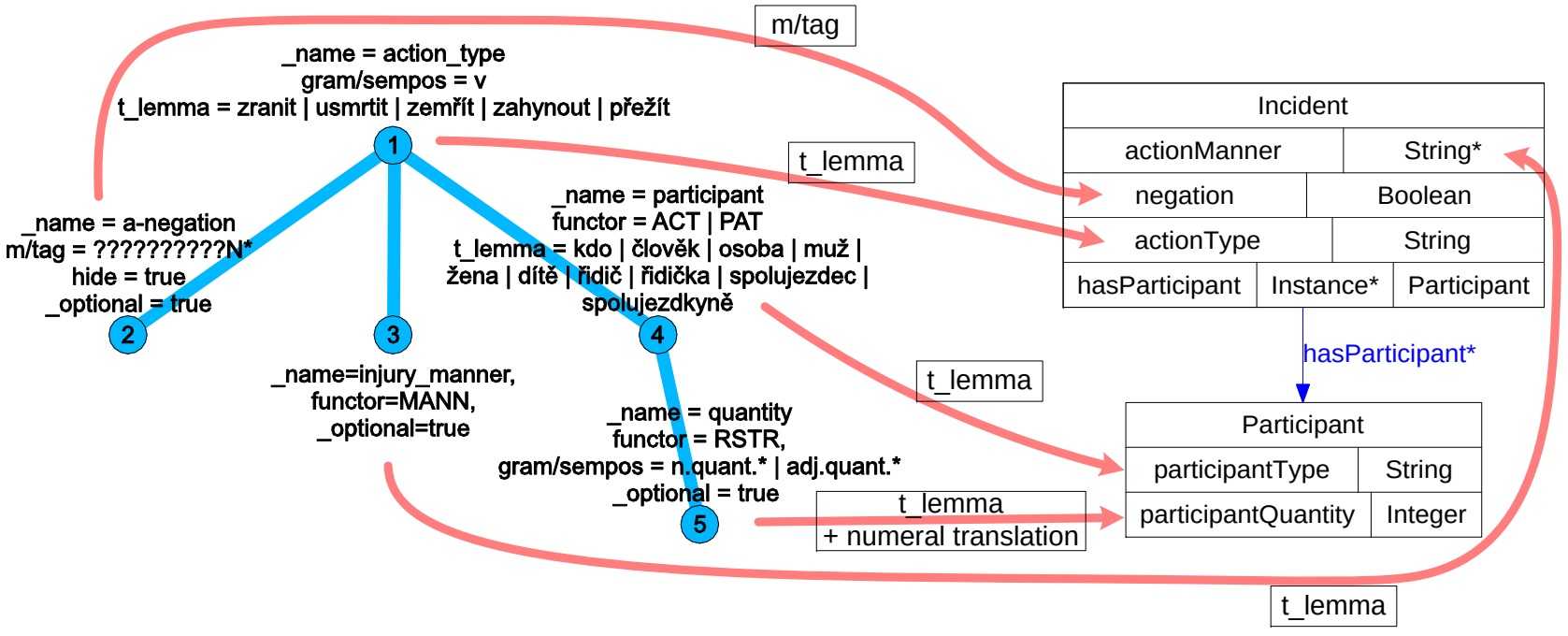
- Tree patterns on shape and nodes (on node attributes).
- Evaluation gives actual matches of particular nodes.
- Names of nodes allow use of references.

Raw data extraction output

```
<QueryMatches>
  <Match root_id="T-vysocina63466.txt-001-p1s4" match_string="2:0,7:3,8:4,11:2">
    <Sentence>
      Při požáru byla jedna osoba lehce zraněna - jednalo se
      o majitele domu, který si vykloubil rameno.
    </Sentence>
    <Data>
      <Value variable_name="action_type" attribute_name="t_lemma">zranit</Value>
      <Value variable_name="injury_manner" attribute_name="t_lemma">lehký</Value>
      <Value variable_name="participant" attribute_name="t_lemma">osoba</Value>
      <Value variable_name="quantity" attribute_name="t_lemma">jeden</Value>
    </Data>
  </Match>
  <Match root_id="T-jihomoravsky49640.txt-001-p1s4" match_string="1:0,13:3,14:4">
    <Sentence>
      Ve zdemolovaném trabantu na místě zemřeli dva muži - 82letý senior
      a další muž, jehož totožnost zjišťují policisté.
    </Sentence>
    <Data>
      <Value variable_name="action_type" attribute_name="t_lemma">zemřít</Value>
      <Value variable_name="participant" attribute_name="t_lemma">muž</Value>
      <Value variable_name="quantity" attribute_name="t_lemma">dva</Value>
    </Data>
  </Match>
  <Match root_id="T-jihomoravsky49736.txt-001-p4s3" match_string="1:0,3:3,7:1">
    <Sentence>Čtyřiatřicetiletý řidič nebyl zraněn.</Sentence>
    <Data>
      <Value variable_name="action_type" attribute_name="t_lemma">zranit</Value>
      <Value variable_name="a-negation" attribute_name="m/tag">VpYS---XR-N A---
      </Value>
      <Value variable_name="participant" attribute_name="t_lemma">řidič</Value>
    </Data>
  </Match>
</QueryMatches>
```

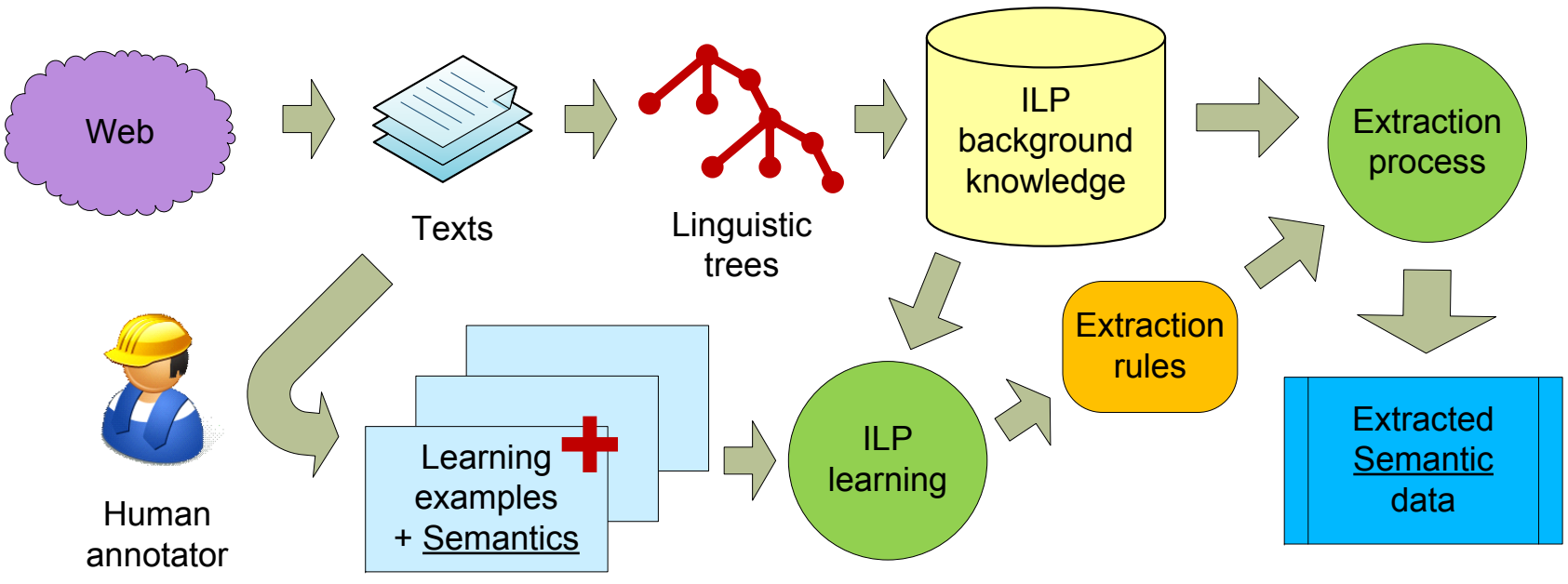
SELECT action_type.t_lemma, a-negation.mtag, injury_manner.t_lemma, participant.t_lemma, quantity.t_lemma **FROM** ****extraction rule****

Semantic interpretation of extraction rules



- Determines how particular values of attributes are used.
- Gives semantics to extraction rule.
- Gives semantics to extracted data.

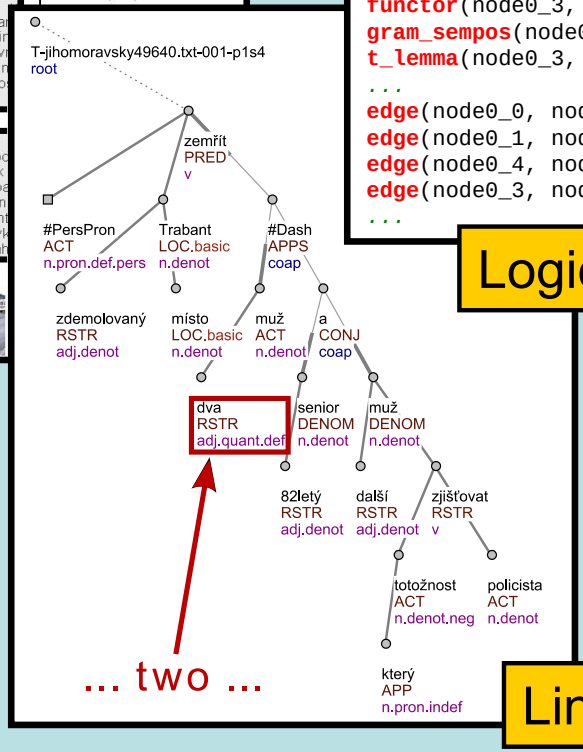
Integration of ILP in our extraction process



- Transformation of trees to logic representation.

Logic representation of linguistic trees

Source web page

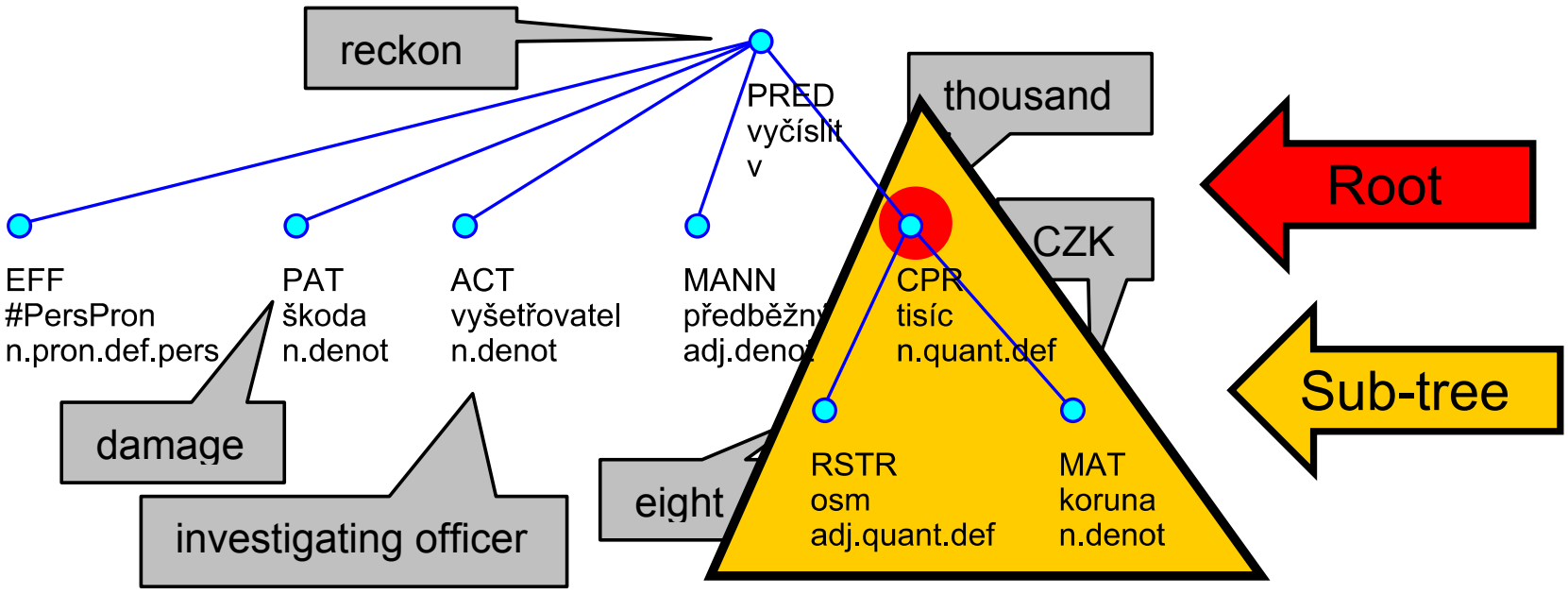


Linguistic trees

```
tree_root(node0_0). node(node0_0).
id(node0_0, t_jihomoravsky49640_txt_001_p1s4).
%%%%%%%% node0_1 %%%%%%%%%%%%%%%%%%
node(node0_1).
functor(node0_1, pred).
gram_sempos(node0_1, v).
t_lemma(node0_1, zemrit).
%%%%%%%% node0_2 %%%%%%%%%%%%%%%%%%
node(node0_2).
functor(node0_2, act).
gram_sempos(node0_2, n_pron_def_pers).
t_lemma(node0_2, x_perspron).
%%%%%%%% node0_3 %%%%%%%%%%%%%%%%%%
node(node0_3). id(node0_3,
functor(node0_3, loc).
gram_sempos(node0_3, n_denot).
t_lemma(node0_3, trabant).
...
edge(node0_0, node0_1). edge(node0_1, node0_2).
edge(node0_1, node0_3). edge(node0_3, node0_4).
edge(node0_4, node0_5). edge(node0_3, node0_6).
edge(node0_3, node0_7). edge(node0_3, node0_8).
...
```

Logic representation

Root/Subtree Preprocessing/Postprocessing (Chunk learning)



…, škodu vyšetřovatel předběžně vyčíslil na osm tisíc korun.

…, investigating officer preliminarily reckoned the damage to be eight thousand Crowns (CZK).

Examples of learned rules, Czech words are translated.

Example

[Rule 1] [Pos cover = 14 Neg cover = 0]

```
damage_root(A) :- lex_rf(B,A), has_sempos(B,'n.quant.def'),
   tDependency(C,B), tDependency(C,D),
   has_t_lemma(D,'investigator').
```

[Rule 2] [Pos cover = 13 Neg cover = 0]

```
damage_root(A) :- lex_rf(B,A), has_functor(B,'TOWH'),
   tDependency(C,B), tDependency(C,D), has_t_lemma(D,'damage').
```

[Rule 1] [Pos cover = 7 Neg cover = 0]

```
injuries(A) :- lex_rf(B,A), has_functor(B,'PAT'),
   has_gender(B,anim), tDependency(B,C), has_t_lemma(C,'injured').
```

[Rule 8] [Pos cover = 6 Neg cover = 0]

```
injuries(A) :- lex_rf(B,A), has_gender(B,anim), tDependency(C,B),
   has_t_lemma(C,'injure'), has_negation(C,neg0).
```

Evaluation results

task/method	matching	missing	excess	overlap	prec.%	recall%	F1.0%
damage/ILP	14	0	7	6	51.85	70.00	59.57
damage/ILP – lenient measures					74.07	100.00	85.11
dam./ILP-roots	16	4	2	0	88.89	80.00	**84.21**
damage/Paum	*20*	*0*	*6*	*0*	*76.92*	*100.00*	*86.96*
injuries/ILP	15	18	11	0	57.69	45.45	**50.85**
injuries/Paum	*25*	*8*	*54*	*0*	*31.65*	*75.76*	*44.64*
inj./Paum-afun	*24*	*9*	*38*	*0*	*38.71*	*72.73*	*50.53*

- 10-fold cross validation
- Two tasks: ‘damage’ and ‘injuries’
- Root/subtree preprocessing/postprocessing used for ‘damage’ task

Summary

- Implemented a system for extraction of semantic information
- Based on third party linguistic tools (TectoMT[2])
- Extraction rules adopted from Netgraph[3] application.
- ILP used for learning rules.
- All methods integrated inside GATE[4].

- Main advantages:
  - Automated selection of learning features
  - “Language independent”
  - Rule based

[2] http://ufal.mff.cuni.cz/tectomt/
[3] http://quest.ms.mff.cuni.cz/netgraph/
[4] http://gate.ac.uk/

Future work

- Use some Knowledge Base (e.g. WordNet).
- Adaptation of this method on other languages.
- Evaluation of the method on other datasets.
- Be able to provide more semantics.
  - e.g. sophisticated semantic interpretation of extracted data