Fuzzy ILP and Semantic Information Extraction from Texts

Jan Dědek

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

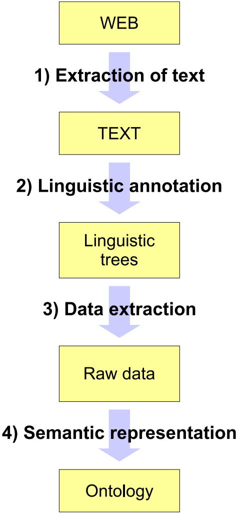
KEG, 15 October 2009, VŠE, Praha

Outline

Introduction to Presented Work

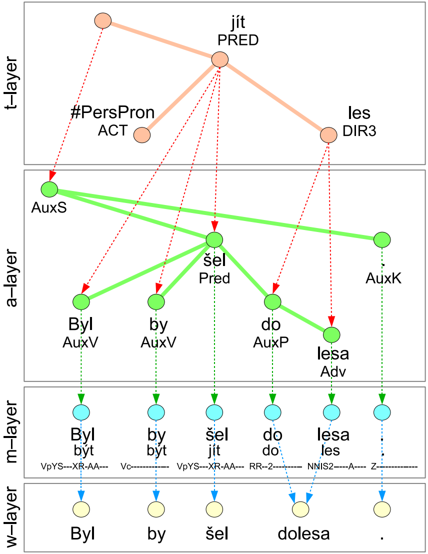
- Extraction of semantic information from texts.
  - In Czech language.
  - Coming from web pages.
- Using of Semantic Web ontologies.
  - RDF, OWL
- Exploiting of linguistic tools.
  - Mainly from the Prague Dependency Treebank project.
  - Experiments with the Czech WordNet.
- Rule based extraction method.
  - Extraction rules $\approx$ tree queries
  - ILP learning of extraction rules

Schema of the extraction process



1. Extraction of text
   - Using RSS feed to download pages.
   - Regular expression to extract text.
2. Linguistic annotation
   - Using chain of 6 linguistic tools (see on next slides).
3. Data extraction
   - Exploitation of linguistic trees.
   - Using extraction rules.
4. Semantic representation of data
   - Ontology needed.
   - Semantic interpretation of rules.
   - Far from finished in current state.

Layers of linguistic annotation in PDT



- Tectogrammatical layer
- Analytical layer
- Morphological layer

Sentence:

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

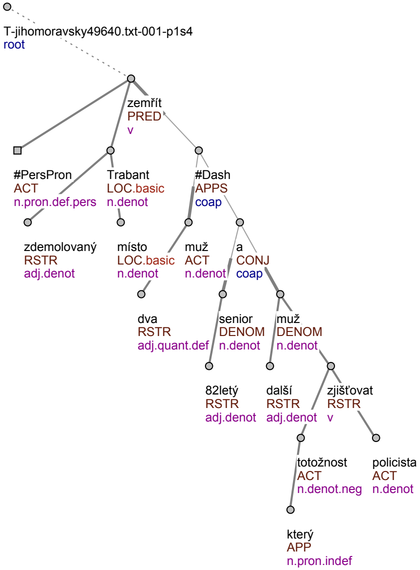
Tools for machine linguistic annotation

Available on the PDT 2.0 CD-ROM

1. Segmentation and tokenization
2. Morphological analysis
3. Morphological tagging
4. Collins' parser – Czech adaptation
5. Analytical function assignment

6. Tectogrammatical analysis
   – Developed by Václav Klimeš

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 – . . .

Example of the web-page with a report of a fire department

Ministerstvo vnitra

>home >navigace >vyhledávání >změna vzhledu

Zpravodajství

Informace z resortu o tom, co se stalo, co se děje i co se připravuje

HZS Jihomoravského kraje

Zubatého 1, 614 00 Brno, telefon 950 630 111,
http://www.firebrno.cz
Zpravodajství v roce 2006

15.05.2007

V trabantu zemřeli dva lidé

K tragické nehodě dnes odpoledne hasiči vyjížděli na silnici z obce Česká do Kuřimi na Brněnsku.

Nehoda byla operačnímu středisku HZS ohlášena ve 13.13 hodin a na místě zasahovala jednotka profesionálních hasičů ze stanice v Tišnově. Jednalo se o čelní srážku autobusu Karosa s vozidlem Trabant 601. Podle dostupných informací trabant jedoucí ve z Brna do Kuřimi zřejmě vyjel do protisměru, kde narazil do linkového autobusu dopravní společnosti ze Žďáru nad Sázavou. Ve zdemolovaném trabantu na místě zemřeli dva muži – 82letý senior a další muž, jehož totožnost zjišťují policisté.

Hasiči udělali na vozidle protipožární opatření a po vyšetření a zadokumentování nehody dopravní policií vrak trabantu zaklesnutý pod autobusem pomocí lana odtrhli. Po odstranění střechy trabantu pak z kabiny vyprostili těla obou mužů. Obě vozidla – trabant i autobus, pak postupně odstranili na kraj vozovky a uvolnili tak jeden jízdní pruh. Únik provozních kapalin nebyl zjištěn. Po 16. hodině pomohli vrak trabantu naložit k odtahu a asistovali při odtažení autobusu. Po úklidu vozovky krátce před 16.30 hod. místo nehody předali policistům a ukončili zásah.

skryj menu

Odkazy

Hasiči

- Generální ředitelství
- hl. m. Praha
- Jihočeský kraj
- Jihomoravský kraj
- Karlovarský kraj
- Královéhradecký kraj
- Liberecký kraj
- Moravskoslezský kraj
- Olomoucký kraj
- Pardubický kraj
- Plzeňský kraj
- Středočeský kraj
- Ústecký kraj
- kraj Vysočina
- Zlínský kraj

Euroskop.cz

V této rubrice Zpravodajství

- Aktualizace stránek
- Archiv zpravodajství
- Bleskové zpravodajství RSS
- Boj proti korupci
- Digitální televize
- Hasiči
- Hlavní zprávy
- Ministerstvo
- Od dopisovatelů (neoficiální)
- Policie
- Regiony
- Servis nejen pro novináře
- Schengenská spolupráce
- WebEditorial

Na našem serveru v jiných rubrikách

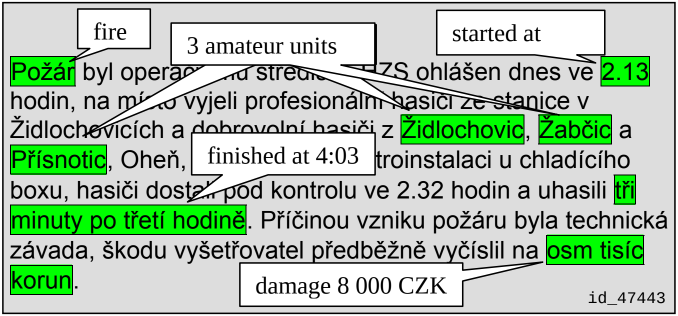
- Aktuality Národního archivu

Domain of our experiments

- Fire-department articles
- Published by The Ministry of Interior of the Czech Republic[1]
- Processed more than 800 articles
  from different regions of Czech Republic
- 1.2 MB of textual data
- Linguistic tools produced 10 MB of annotations,
  run time 3.5 hours
- Extracting information about injured and killed people
- 470 matches of the extraction rule,
  200 numeric values of quantity (described later)

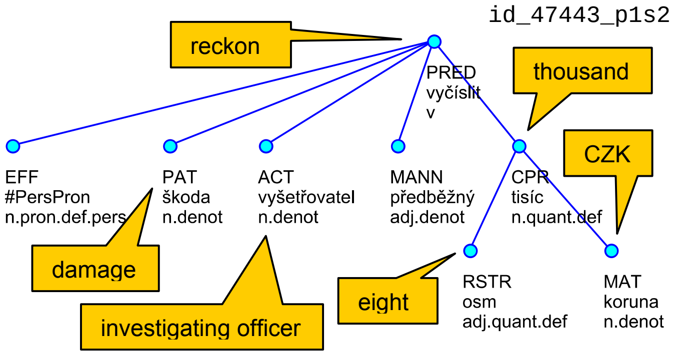
[1] `http://www.mvcr.cz/rss/regionhzs.html`

Example of processed text



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

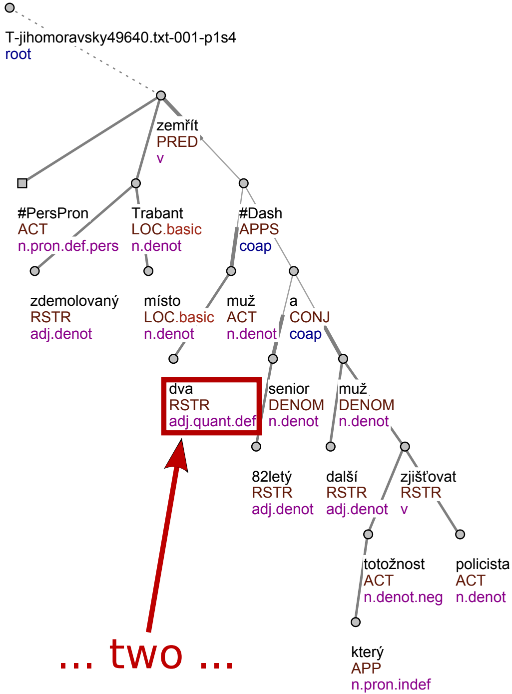
Example of a linguistic tree



…, škodu vyšetřovatel předběžně vyčíslil na osm tisíc korun.

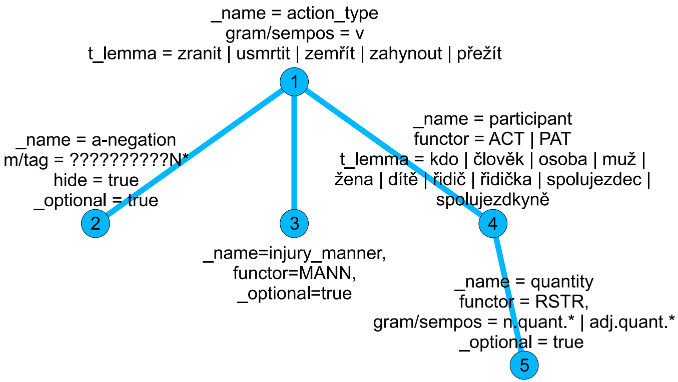
…, investigating officer preliminarily reckoned the damage to be 8 000 CZK.

- Our IE method uses tree queries (tree patterns)



- How to extract the information about two dead people?

Extraction rules – Netgraph queries



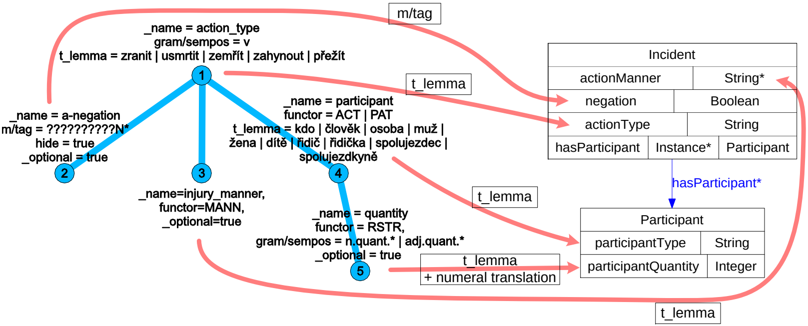
- Tree patterns on shape and nodes (on node attributes).
- Evaluation gives actual matches of particular nodes.
- Names of nodes allow use of references.

Raw data extraction output


```xml
<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-NA---
      </Value>
      <Value variable_name="participant" attribute_name="t_lemma">řidič</Value>
    </Data>
  </Match>
</QueryMatches>
```

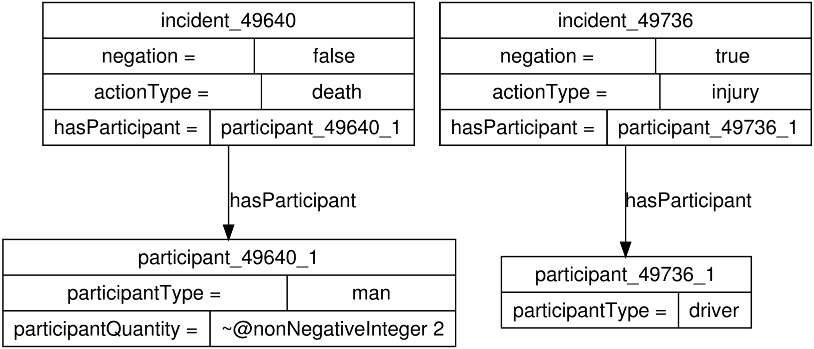

SELECT action_type.t_lemma, a-negation.mtag, injury_manner.t_lemma, participant.t_lemma, quantity.t_lemma **FROM** ****extraction rule****

Semantic interpretation of extraction rules



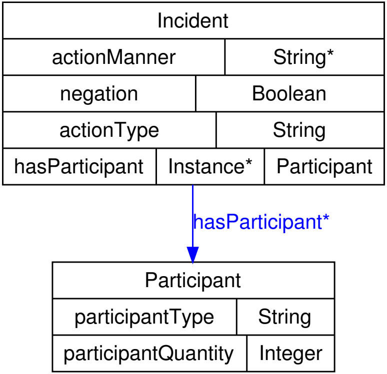
- Determines how particular values of attributes are used.
- Gives semantics to extraction rule.
- Gives semantics to extracted data.

Semantic data output



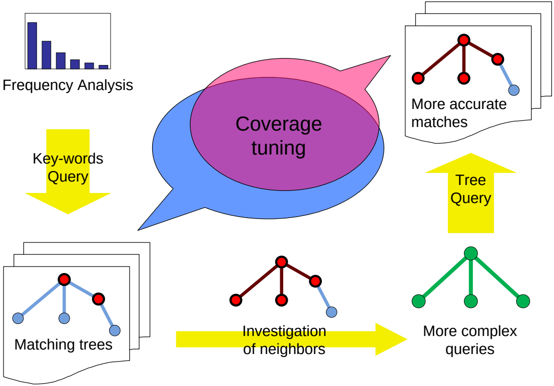
- Two instances of two ontology classes.

The experimental ontology



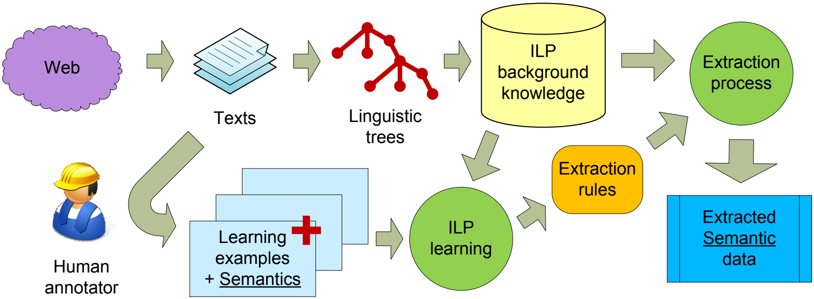
- Two classes
  - Incident and Participant
- One object property relation
  - hasParticipant
- Five datatype property relations
  - actionManner (light or heavy injury)
  - negation
  - actionType (injury or death)
  - participantType (man, woman, driver, etc.)
  - participantQuantity

Design of extraction rules – iterative process



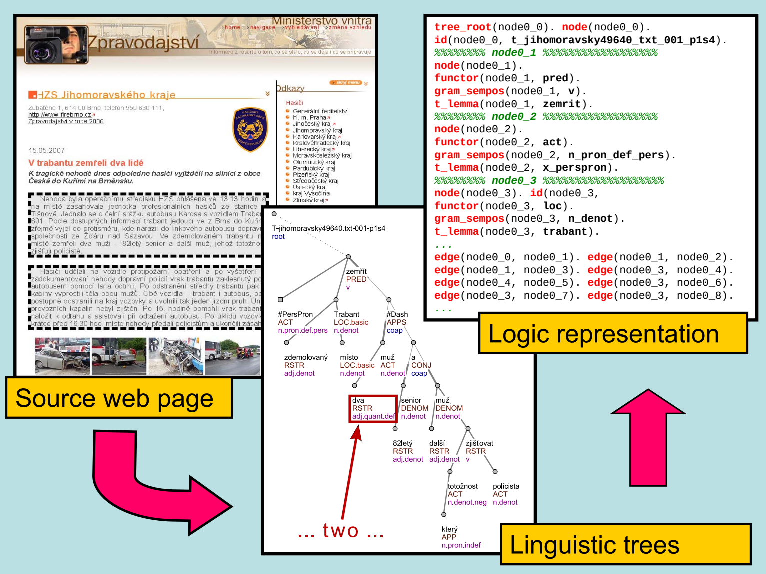
1. Frequency analysis → representative key-words.
2. Investigating of matching trees → tuning of tree query.
3. Complexity of the query ≅ complexity of extracted data.

Integration of ILP in our extraction process



- Transformation of trees to logic representation.
- Today: just first promising experiments.

Logic representation of linguistic trees



First promising results :-)

Example

```
contains_num_injured(A) :- t_lemma(A,1).
contains_num_injured(A) :- t_lemma(A,2).
contains_num_injured(A) :- t_lemma(A,23).
contains_num_injured(A) :- edge(A,B), m_form(B,jeden).
contains_num_injured(A) :- edge(A,B), m_tag(B,cn_s1__________).
contains_num_injured(A) :- edge(B,A), functor(B,conj).
contains_num_injured(A) :- edge(B,A), t_lemma(B,dite).
contains_num_injured(A) :- edge(B,A), t_lemma(B,muz).
contains_num_injured(A) :- edge(B,A), edge(B,C), m_tag14(C,1).
contains_num_injured(A) :- edge(B,A), edge(B,C), t_lemma(C,tezky).
contains_num_injured(A) :- edge(B,A), edge(B,C), t_lemma(C,nasledek).
contains_num_injured(A) :- edge(A,B), edge(C,A), m_tag4(B,1), functor(C,pat).
contains_num_injured(A) :- edge(A,B), edge(C,A), functor(C,act), a_afun(B,sb).
contains_num_injured(A) :- edge(B,A), edge(C,B), edge(C,D), t_lemma(D,vloni).
contains_num_injured(A) :- edge(B,A), edge(C,B), t_lemma(B,osoba), t_lemma(C,zranit).
contains_num_injured(A) :- edge(B,A), edge(C,B), t_lemma(B,osoba), t_lemma(C,zemrit).
contains_num_injured(A) :- edge(B,A), edge(C,B), functor(B,act), edge(C,D),
                           a_afun(D,obj).
contains_num_injured(A) :- edge(B,A), edge(C,B), t_lemma(B,osoba), edge(C,D), edge(D,E),
                           functor(D,twhen).
contains_num_injured(A) :- edge(B,A), t_lemma(A,tri), edge(B,C), edge(D,B), edge(E,D),
                           m_tag2(C,m).
```

1. Introduction
   - Our Information Extraction System
   - Linguistics we have used.
   - Domain of fire-department articles
2. Our Information Extraction Method
   - Manually created rules
   - Learning of rules
3. **Fuzzy ILP**
   - Introd. example, theory, architecture and an experiment
   - Fuzzy ILP Implementation
   - Evaluation and Conclusion
4. Conclusion

ILP Example

Types of ground variables

```
animal(dog). animal(dolphin) ... animal(penguin).
class(mammal). class(fish). class(reptile). class(bird).
covering(hair). covering(none). covering(scales).
habitat(land). habitat(water). habitat(air).
```

Background knowledge

```
has_covering(dog, hair). has_covering(crocodile, scales).
has_legs(dog,4). ... has_legs(penguin, 2). etc.
has_milk(dog). ... has_milk(platypus). etc.
homeothermic(dog). ... homeothermic(penguin). etc.
habitat(dog, land). ... habitat(penguin, water). etc.
has_eggs(platypus). ... has_eggs(eagle). etc.
has_gills(trout). ... has_gills(eel). etc.
```

ILP Example

Positive examples

```
class(lizard, reptile).
class(trout, fish).
class(bat, mammal).
```

Negative examples

```
class(trout, mammal).
class(herring, mammal).
class(platypus, reptile).
```

Induced rules

```
class(A,reptile) :- has_covering(A,scales),
                    has_legs(A,4).
class(A,mammal) :- homeothermic(A), has_milk(A).
class(A,fish) :- has_legs(A,0), has_eggs(A).
class(A,reptile) :- has_covering(A,scales),
                    habitat(A,land).
class(A,bird) :- has_covering(A,feathers).
```

Classical ILP and Fuzzy ILP principles

- Learning examples $E = P \cup N$ (Positive and Negative)
- Background knowledge B
- ILP task – to find hypothesis H such that:

$$(\forall e \in P)(B \cup H \models e) \;\;\&\;\; (\forall n \in N)(B \cup H \not\models n).$$

- Fuzzy learning examples $\mathcal{E} : E \longrightarrow [0, 1]$
- Fuzzy background knowledge $\mathcal{B} : B \longrightarrow [0, 1]$
- Fuzzy ILP task – to find hyp. $\mathcal{H} : H \longrightarrow [0, 1]$ such that:

$$(\forall e_1, e_2 \in E)(\forall \mathcal{M})(\mathcal{M} \models_f \mathcal{B} \cup \mathcal{H}) \;:\; \mathcal{E}(e_1) > \mathcal{E}(e_2) \Rightarrow \|e_1\|_{\mathcal{M}} \geq \|e_2\|_{\mathcal{M}}$$

Generalized Annotated Programs

- Fuzzy ILP is equivalent to Induction of Generalized Annotated Programs[2]
- For implementation we use GAP or strictly speaking: *Definite Logic Programs with monotonicity axioms* (also equivalent)
- Basic paradigm: deal with values as with degrees.
  - We don't have to normalize values, they order is enough.

- For example with monotonicity axioms we can use rule:

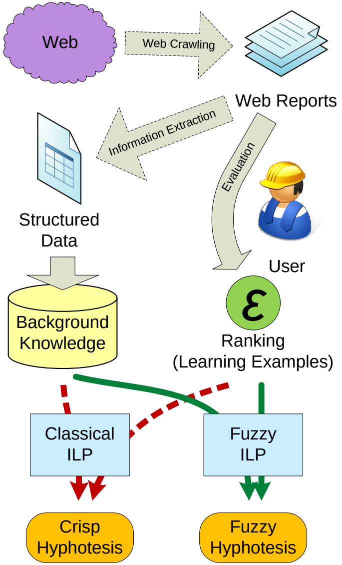
  `serious(A, 4)` ← `fatalities(A, 10).`

  and from the fact `fatalities(id_123, 1000)` deduce

  `serious_alt(id_123, 4).`

[2]See in S. Krajci, R. Lencses and P. Vojtas: "A comparison of fuzzy and annotated logic programming", Fuzzy Sets and Systems, vol.144, pp.173–192, 2004.

Schema of the whole system



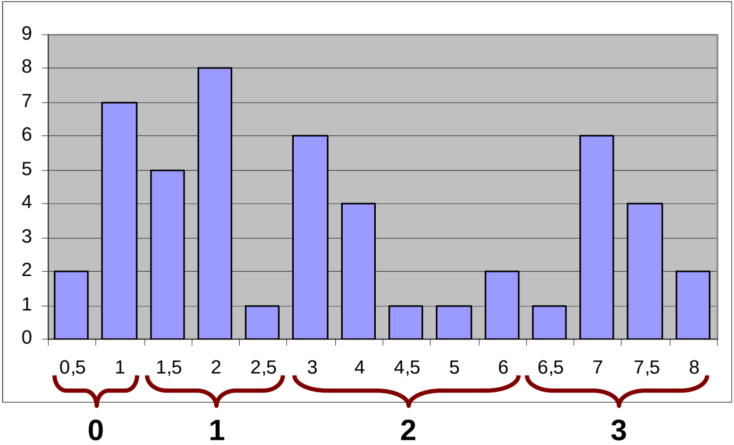
1. Web Crawling
2. Information Extraction and User Evaluation
3. Logic representation
   - Construction of background knowledge
   - Construction of learning examples
4. ILP Learning
   - Crisp
   - Fuzzy
5. Comparison of results

Accident attributes

attribute name	distinct values	missing values	monotonic
size (of file)	49	0	yes
type (of accident)	3	0	no
damage	18	30	yes
dur_minutes	30	17	yes
fatalities	4	0	yes
injuries	5	0	yes
cars	5	0	yes
amateur_units	7	1	yes
profesional_units	6	1	yes
pipes	7	8	yes
lather	3	2	yes
aqualung	3	3	yes
fan	3	2	yes
ranking	14	0	yes

- Information that could be extracted.
- Missing values.
- Almost all attributes are numeric.
  - So monotonic
  - This will be used for “fuzzyfication”
- Artificial target attribute seriousness ranking.

Histogram of the seriousness ranking attribute



- 14 different values, range 0.5 – 8
- Divided into four approximately equipotent groups.

1. **Introduction**
   - Our Information Extraction System
   - Linguistics we have used.
   - Domain of fire-department articles
2. **Our Information Extraction Method**
   - Manually created rules
   - Learning of rules
3. **Fuzzy ILP**
   - Introd. example, theory, architecture and an experiment
   - Fuzzy ILP Implementation
   - Evaluation and Conclusion
4. **Conclusion**

Essential difference between learning examples

Crisp learning examples

```
serious_2(id_47443). %positive

serious_0(id_47443). %negative
serious_1(id_47443). %negative
serious_3(id_47443). %negative
```

Monotonized learning examples

```
serious_atl_0(id_47443). %positive
serious_atl_1(id_47443). %positive
serious_atl_2(id_47443). %positive

serious_atl_3(id_47443). %negative
```

For one evidence (occurrence):

- Crisp:
  Always one positive and three negative learning examples
- Monotonized:
  Up to the observed degree positive, the rest negative.

Monotonization of attributes

damage → damage_atl

```
damage_atl(ID,N) :- %unknown values
        damage(ID,N), not(integer(N)).
damage_atl(ID,N) :- %numeric values
        damage(ID,N2), integer(N2),
        damage(N), integer(N), N2>=N.
```

- We infer all lower values as sufficient.
- Treatment of unknown values.
- Negation as failure.

```
serious_0(A):-dur_minutes(A,8).
serious_0(A):-type(A,fire),pipes(A,0).
serious_0(A):-fatalities(A,0),pipes(A,1),lather(A,0).
serious_1(A):-amateur_units(A,1).
serious_1(A):-amateur_units(A,0),pipes(A,2),aqualung(A,1).
serious_1(A):-damage(A,300000).
serious_1(A):-damage(A,unknown),type(A,fire),prof_units(A,1).
serious_1(A):-dur_minutes(A,unknown), fatalities(A,0), cars(A,1).
serious_2(A):-lather(A,unknown).
serious_2(A):-lather(A,0), aqualung(A,1), fan(A,0).
serious_2(A):-amateur_units(A,2),prof_units(A,2).
serious_2(A):-dur_minutes(A,unknown),injuries(A,2).
serious_3(A):-fatalities(A,1).
serious_3(A):-fatalities(A,2).
serious_3(A):-injuries(A,2), cars(A,2).
serious_3(A):-pipes(A,4).
```

- Crisp hypothesis

```
serious_atl_0(A).
serious_atl_1(A):-injuries_atl(A,1).
serious_atl_1(A):-lather_atl(A,1).
serious_atl_1(A):-pipes_atl(A,3).
serious_atl_1(A):-dur_minutes_atl(A,unknown).
serious_atl_1(A):-size_atl(A,764),pipes_atl(A,1).
serious_atl_1(A):-damage_atl(A,8000),amateur_units_atl(A,3).
serious_atl_1(A):-type(A,car_accident).
serious_atl_1(A):-pipes_atl(A,unknown), randomized_order_atl(A,35).
serious_atl_2(A):-pipes_atl(A,3), aqualung_atl(A,1).
serious_atl_2(A):-type(A,car_accident), cars_atl(A,2),prof_units_atl(A,2).
serious_atl_2(A):-injuries_atl(A,1),prof_units_atl(A,3),fan_atl(A,0).
serious_atl_2(A):-type(A,other), aqualung_atl(A,1).
serious_atl_2(A):-dur_minutes_atl(A,59), pipes_atl(A,3).
serious_atl_2(A):-injuries_atl(A,2),cars_atl(A,2).
serious_atl_2(A):-fatalities_atl(A,1).
serious_atl_3(A):-fatalities_atl(A,1).
serious_atl_3(A):-dur_minutes_atl(A,unknown),pipes_atl(A,3).
```

- Monotonized hypothesis
  - Monotonicity axioms
  - Monotonized learning examples

Evaluation and Comparison of Results

			Raw ILP	Monot. ILP
Monot. test set		TP:	42	57
positive:	64	FP:	7	6
negative:	36	Precision:	0,857	0,905
sum:	100	Recall:	0,656	0,891
		F-measure:	0,743	0,898
Crisp test set		TP:	12	15
positive:	25	FP:	13	10
negative:	75	Precision:	0,480	0,600
sum:	100	Recall:	0,480	0,600
		F-measure:	0,480	0,600

- Rules evaluated on both testing sets.
  - By use of conversion predicates (next slide)
- Monotonized rules better in both cases.
- Even better than other classifiers (Znalosti 2010).

Conversion of Results

crisp → monotone

```
serious_2(ID) :- serious_atl_2(ID),
                 not(serious_atl_3(ID)).
```

monotone → crisp

```
serious_atl_0(ID) :- serious_2(ID).
serious_atl_1(ID) :- serious_2(ID).
serious_atl_2(ID) :- serious_2(ID).
```

Summary

- Proposed a system for extraction of semantic information
- Based on linguistic tools for automatic text annotation
- Extraction rules adopted from Netgraph application.
- ILP used for learning rules.

- Our future research will concentrate on:
    - Learning of extraction rules.
    - Extension of the method with WordNet technology.
    - Adaptation of this method on other languages.
    - Evaluation of the method.