Organizer: Institute of Formal and Applied Linguistics,
Faculty of Mathematics and Physics, Charles University in Prague,
Malostranské nám. 25, 118 00 Prague, Czech Republic,
`http://ufal.mff.cuni.cz/`

Prague Treebanking for Everyone: A two-day tutorial

Tutorial Notes

November 28-29, 2006
Prague, Czech Republic

The tutorial is supported by:

Faculty of Mathematics and Physics
Charles University in Prague
Ke Karlovu 3, 121 16 Praha 2
Czech Republic
`http://www.mff.cuni.cz/`

Czech Society for Cybernetics
and Informatics
Pod Vodárenskou věží 2
182 07 Praha 8 – Libeň
Czech Republic
`http://www.cski.cz/`

Central European Initiative
CEI - Executive Secretariat
Via Genova, 9
34121 Trieste, Italy
`http://www.ceinet.org/`

Open Society Institute
400 West 59th Street
New York, NY 10019
USA
`http://www.soros.org/`

Preface

Traveling to Prague in autumn 2006 is a great opportunity to enjoy events and meet people behind them which have to do something with computational linguistics. The Institute of Formal and Applied Linguistics organizes the following international events at this time: the Vilém Mathesius Centre Lecture Series 21, the 5th international “Treebanks and Linguistic Theories” conference and, last but not least, the “**Prague Treebanking for Everyone**” tutorial, the notes of which you are visiting now.

The tutorial introduces the Prague Dependency Treebank project, which aims at a complex (mainly) manual annotation of a substantial amount of naturally occurring sentences in continuous Czech texts. The Prague Dependency Treebank has three levels of annotation: morphological, analytical (describing surface syntax in a dependency fashion) and tectogrammatical, which combines syntax and sentence semantics into a language meaning representation, keeping the dependency structure as the core of the annotation structure but adding basic coreferential links, topic/focus annotation, and a detailed semantic labeling of every sentence unit. Several other treebanks having originated in Prague are introduced as well (the Czech Academic Corpus, the Prague Czech-English Dependency Treebank, and the Prague Arabic Dependency Treebank). In addition to the data, all the treebank and data processing tools are discussed in details.

The tutorial is given by nine different speakers from the host institute. Such high number (representing in fact a much higher number of the Prague “treebankers”) reflects the fact that collecting a huge bank of trees is unimaginable without a real teamwork with a strong leadership building on a solid theoretical basis. Fortunately, in case of all Prague treebanks these presumptions are fulfilled.

The speakers’ notes in the tutorial material are arranged in the same order as they are presented during the tutorial. They are accompanied by three colorful solid cards to help the users to become more familiar with morphological tags, analytical functions and tectogrammatical attributes used in the Prague treebanks. Together with them two cd-roms are provided: the PDT 2.0 cd-rom and the CAC 1.0 cd-rom consisting of everything what is included in their original distributions (by the Linguistic Data Consortium and the Charles University Press, respectively) except for the full data.

We wish you to have a wonderful time in Prague!

Jan Hajič
Eva Hajičová
Barbora Hladká

List of Speakers

Name	e-mail address
Jan Hajič	`hajic@ufal.mff.cuni.cz`
Eva Hajičová	`hajicova@ufal.mff.cuni.cz`
Jaroslava Hlaváčová	`hlava@ufal.mff.cuni.cz`
Ondřej Kučera	`kucera@ufal.mff.cuni.cz`
Jiří Mírovský	`mirovsky@ufal.mff.cuni.cz`
Petr Pajas	`pajas@ufal.mff.cuni.cz`
Otakar Smrž	`smrz@ufal.mff.cuni.cz`
Jan Štěpánek	`stepanek@ufal.mff.cuni.cz`
Zdeněk Žabokrtský	`zabokrtsky@ufal.mff.cuni.cz`

Time Schedule

NOVEMBER 28, 2006
Tuesday

09:30-11:00	**Part 1** DATA: The Prague Dependency Treebank and the Czech Academic Corpus (*Jan Hajič*) • Introduction • Morphology
11:30-13:00	**Part 2** DATA (continued): PDT (*Jan Hajič*) • Surface Dependency Syntax • "Deep" (Tectogrammatical) Syntax
13:00-14:30	Lunch
14:30-16:00	**Part 3** DATA (continued): PDT • Grammatemes (*Zdeněk Žabokrtský*) • "Deep" Syntax: topic/focus and deep word order (*Eva Hajičová*) • Coreference (*Eva Hajičová*)
16:30-18:00	**Part 4** DATA (continued): PDT: Valency (*Jan Hajič*)

Time Schedule (continued)

NOVEMBER 29, 2006
Wednesday

09:30-11:00	**Part 5** TOOLS • Annotation editors • m-layer: LAW (*Jaroslava Hlaváčová*) • [at]-layer, valency lexicon: TrEd (*Jan Štěpánek*) • Browsers and viewers • m-layer: Bonito (*Jaroslava Hlaváčová*) • [at]-layer: Netgraph (*Jiří Mírovský*)
11:30-12:30	**Part 6** DATA: The Prague Mark-up Language (*Petr Pajas*)
12:30-14:30	**Lunch**
14:30-16:00	**Part 7** TOOLS (continued): • Automatic processing of data (*Jan Štěpánek*) • STYX - an electronic exercise book of Czech (*Ondřej Kučera*)
16:30-18:00	**Part 8** DATA: More Prague Treebanks • Prague Czech-English Dependency Treebank (*Jan Hajič*) • Prague Arabic Dependency Treebank (*Otakar Smrž*)

Part 1

DATA:
The Prague Dependency Treebank and the Czech Academic Corpus

The Prague Dependency Treebank and Valency Annotation (part 1)

Jan Hajič
Institute of Formal and Applied Linguistics
School of Computer Science
Faculty of Mathematics and Physics
Charles University, Prague
Czech Republic

Tutorial Outline – “the Data”

- (1) The Prague Dependency Treebank (PDT)
    - Introduction, token level, morphology
    - “Physical” markup / intro
- (2) The Syntactic Annotation of the PDT
    - Surface syntactic annotation
    - “Deep” Syntactic Structure, Valency (intro)
- [(3) Topic/focus, Coreference, Grammatemes]
- (4) Tectogrammatical Annotation & Valency Lexicon
    - Verbs and Nouns: Relating Form, Syntax and Semantics
    - Linking the Corpus and the Lexicon
    - Using the annotated corpus – further research and tools

---

Prague Dependency Treebank Intro, tokens, morphology (p. 1)

- Introduction to the Prague Dependency Treebank family of projects
- Text base, tokenization, sentence boundaries
- Morphology
    - Lexicon, lemmatization
    - Inflection morphology
    - Tagset
    - Manual annotation process

The Prague Dependency Treebank Project (Czech Treebank)

- 1996-2005-...
    - 1998 PDT v. 0.5 released (JHU workshop)
        - 400k words annotated, unchecked
    - 2001 PDT 1.0 released (LDC):
        - 1.3MW annotated, morphology & surface syntax
    - 2005 PDT 2.0 release planned
        - 0.8MW annotated (50k sentences)
        - the “tectogrammatical layer”
            - underlying (deep) syntax

Related Projects (Treebanks)

- Prague Czech-English Dependency Treebank
  - WSJ portion of PTB, translated to Czech
  - automatically analyzed
    - English side (PTB), too
- Prague Arabic Dependency Treebank
  - apply same representation to annotation of Arabic
  - suface syntax so far
- Both have been published in 2004 (LDC)
- Czech Academic Corpus v. 1.0 (2006)
  - Conversion of 70s' style annotation to PDT style (0.5 mil.)

PDT (Czech) Data

- 4 sources:
  - Lidové noviny (daily newspaper, incl. extra sections)
  - DNES (Mladá fronta Dnes) (daily newspaper)
  - Vesmír (popular science magazine, monthly)
  - Českomoravský Profit (economical journal, weekly)
- Full articles selected
  - article ~ <u>DOCUMENT</u> (basic corpus unit)
- Time period: 1990-1995
- 1.8 million tokens (~110 thousand sentences)

PDT Annotation Layers

PDT 1.0 (2001)

PDT 2.0 (2006)

- L0 (w) Words (tokens)
  - automatic segmentation and markup only
- L1 (m) Morphology
  - Tag (full morphology, 13 categories), lemma
- L2 (a) Analytical layer (surface syntax)
  - Dependency, analytical dependency function
- L3 (t) Tectogrammatical layer ("deep" syntax)
  - Dependency, functor (detailed), grammatemes, ellipsis solution, coreference, topic/focus (deep word order), valency lexicon

---

Tokenization, Segmentation, Sentence Breaks (L0, w-layer)

- Basic Principles
  - Fully automatic
    - Will have to be the same for the manually annotated part as well as for other plain-text data
  - No access to any linguistic knowledge
    - …beyond, say, <u>really</u> fail-safe lists of certain types of abbreviations, language identification, coding scheme, and letter classification (upper/lower/…)
  - Standard output markup
    - unified coding scheme (today, Unicode in most cases)

Tokenization

- Words
  - What is a word? (word boundaries)
    - Treatment of hyphens, apostrophes, periods,…
    - Numbers w/digits (normalization)
      - “periods”, thousand separators
      - Types of numbers (?)
        - cardinal, ordinal, money, SSN, tel/fax/…, dates, ...
    - Mixed letters and digits
  - Rule of thumb:
    - Split whenever there is the slightest doubt!

Tokenization

- Capitalization
  - Main issues (the “true case”):
    - Names (not identified yet!)
    - Start of sentence (don’t know it yet either!)
    - Typographical conventions (unmarked in most cases)
  - Nontrivial
    - Headings
  - Rule of thumb:
    - don’t solve it (yet), just keep it & possibly mark it

(No) Segmentation, I

- Segmentation ~ (for us) splitting “inside” words (“between two letters”)
  - examples (not segmented in PDT):
    - elektro|technický (*electrotechnical*)
    - bílo|červeno|modrý (*white-red-blue*)
    - tisíci|hlavý (*one-thousand-headed*)
    - polo|šílený (*half-mad*)
    - na|č = na co (onto what, contraction (~ isn’t))
    - pracoval|s = pracoval jsi (you have worked, ~ y’know)
    - za|č|s = za co jsi (for what you have <verb>)

(No) Segmentation, II

- Ambiguity
  - přenos:
    - přenos - *transmission*
    - přeno|s - you-have-been argued-with
  - a few others
- However: it is not very frequent (Cz, En, Ar) ⟶
  - can be handled by expanded dictionary & tagset design
  - therefore <u>no segmentation</u> (of this kind)!

Sentence Boundaries

- Chicken and egg problem:
  - To analyze a text linguistically, we need to know sentence boundaries…
    - but…
  - To know sentence boundaries, we would need to have the text linguistically analyzed.
- Solution:
  - Do something good enough in most cases
    - …maybe redo it later in the manually annotated part

PDT Annotation Layers

- L0 (w) Words (tokens)
  - automatic segmentation and markup only
- L1 (m) Morphology
  - Tag (full morphology, 13 categories), lemma
- L2 (a) Analytical layer (surface syntax)
  - Dependency, analytical dependency function
- L3 (t) Tectogrammatical layer (“deep” syntax)
  - Dependency, functor (detailed), grammatemes, ellipsis solution, coreference, topic/focus (deep word order), valency lexicon

---

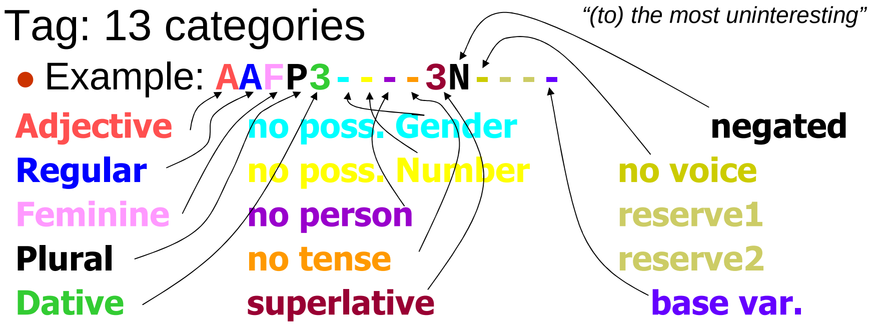
Layer 1 (m-layer): Morphology

- Prerequisites for the manual annotation process:
  - Tokenized data
  - Annotation guidelines
  - Annotation tool
    - Manual decision making support
    - Offline (or online) morphological analyzer
  - Quality checking tool
  - Process description
- Results (manually annotated data) to be used for...
  - tagger training, linguistic research, basis for further annotation, ...

Morphological Attributes

- Tag: 13 categories
  - Example: **AAFP3----3N----**

Ex.: nejnezajímavějším
“(to) the most uninteresting”



Adjective
Regular
Feminine
Plural
Dative
no poss. Gender
no poss. Number
no person
no tense
superlative
negated
no voice
reserve1
reserve2
base var.

- Lemma: POS-unique identifier

Books/verb -> book-1, went -> go, to/prep. -> to-1

Morphological Tagset

- 13 categories, 4452 plausible tags (combinations):

Category	# of values	Example(s)
POS	10	N (noun), Z (punctuation)
SUBPOS	75	P (personal pron.), U (possessive adj.)
GENDER	8	I (masc. inanimate), X (any), - (N.A)
NUMBER	4	P (plural), D (dual)
CASE	9	1 (nominative), 6 (locative)
POSSGENDER	4	M (masc. animate), F (feminine)
POSSNUMBER	3	S (singular), P (plural)
PERSON	5	1 (first), ...
TENSE	4	P (present), M (past)
GRADE	5	3 (superlative)
NEGATION	3	A (affirmative), N (negative)
VOICE	3	A (active), P (passive)
VAR	11	1 (1st variant), 6 (colloq. style), 8 (abbrev.)

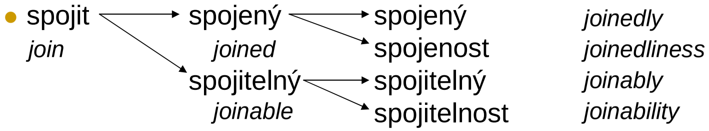
Morphological Analysis

- Formally: MA: $A^+ \rightarrow Pow(L \times T)$
  - MA(f) = { [l,t] };
    - $f \in A^+$ (the token),
    - $l \in L$ (lemma),
    - $t \in T$ (tag)
- tokens taken in isolation
- no attempt to solve e.g. auxiliaries vs. full verbs
- Ex.: MA(“má“) = { [mít,VB-S---3P-AA---], *lit. “to have”*

  lit. “has”,”my” [můj,PSFS1-S1------1], *lit. “my”*
  [můj,PSFS5-S1------1],
  [můj,PSNP1-S1------1],
  [můj,PSNP4-S1------1],
  [můj,PSNP5-S1------1] }

Morphological Analysis: Implementation

- Dictionary-based
  - covers 800kW (lemmas), ~ 20 mil. forms (w/tag)
- C code implementation
  - standard (regular) derivations on-the-fly; ex.:

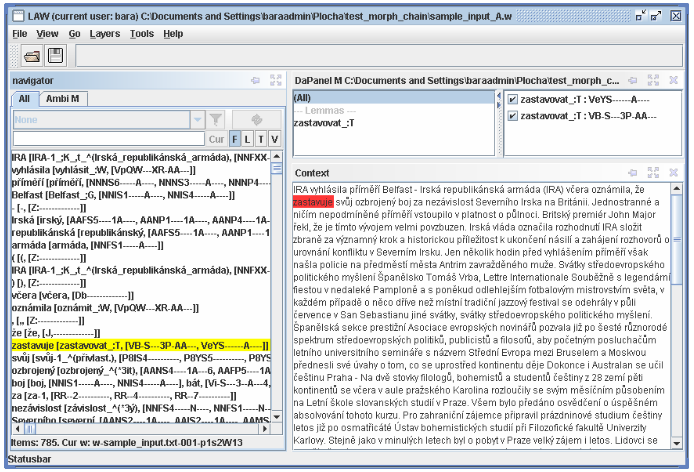


spojit → spojený → spojený *joinedly*
join *joined* → spojenost *joinedliness*
spojit → spojitelný → spojitelný *joinably*
joinable → spojitelnost *joinability*

  - irregular forms listed in dictionary (w/tags)
  - no phonological processing (concatenation only)
  - grammatical prefixes only: negation, superlative

The Morphological Annotation Tool (LAW)

See p. 4
(J. Hlaváčová)



LAW (current user: bara) C:\Documents and Settings\baraadmin\Plocha\test_morph_chain\sample_input_A.w

```
IRA [IRA-1_;K_,t_^(Irská_republikánská_armáda), [NNFXX-
vyhlásila [vyhlásit_:W, [VpQW---XR-AA---]]
příměří [příměří, [NNNS6-----A----, NNNS3-----A----, NNNP4----
Belfast [Belfast_;G, [NNIS1-----A----, NNIS4-----A----]]
- [-, [Z:-------------]]
Irská [irský, [AAFS5----1A----, AANP1----1A----, AANP4----1A-
republikánská [republikánský, [AAFS5----1A----, AANP1----1
armáda [armáda, [NNFS1-----A----]]
( [(, [Z:-------------]]
IRA [IRA-1_;K_,t_^(Irská_republikánská_armáda), [NNFXX-
) [), [Z:-------------]]
včera [včera, [Db-------------]]
oznámila [oznámit_:W, [VpQW---XR-AA---]]
, [,, [Z:-------------]]
že [že, [J,-------------]]
zastavuje [zastavovat_:T, [VB-S---3P-AA---, VeYS------A----]]
svůj [svůj-1_^(přivlast.), [P8IS4----------, P8YS5----------, P8YS
ozbrojený [ozbrojený_^(*3it), [AANS4----1A---6, AAFP5----1A
boj [boj, [NNIS1-----A----, NNIS4-----A----], bát, [Vi-S---3--A---4,
za [za-1, [RR--2----------, RR--4----------, RR--7----------]]
nezávislost [závislost_^(*3ý), [NNFS4-----N----, NNFS1-----N--
Severního [severní, [AANS2----1A----, AAIS2----1A----, AAMS
Items: 785. Cur w: w-sample_input.txt-001-p1s2W13
```

DaPanel M C:\Documents and Settings\baraadmin\Plocha\test_morph_c...

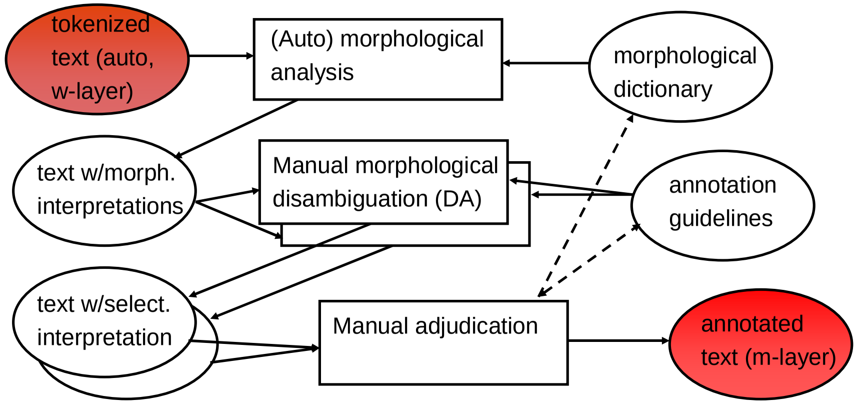
zastavovat_:T

☑ zastavovat_:T : VeYS------A----
☑ zastavovat_:T : VB-S---3P-AA---

IRA vyhlásila příměří Belfast - Irská republikánská armáda (IRA) včera oznámila, že zastavuje svůj ozbrojený boj za nezávislost Severního Irska na Británii. Jednostranné a ničím nepodmíněné příměří vstoupilo v platnost o půlnoci. Britský premiér John Major řekl, že je tímto vývojem velmi povzbuzen. Irská vláda označila rozhodnutí IRA složit zbraně za významný krok a historickou příležitost k ukončení násilí a zahájení rozhovorů o urovnání konfliktu v Severním Irsku. Jen několik hodin před vyhlášením příměří však našla policie na předměstí města Antrim zavražděného muže. Svátky středoevropského politického myšlení Španělsko Tomáš Vrba, Lettre Internationale Souběžně s legendární fiestou v nedaleké Pamploně a s poněkud odlehlejším fotbalovým mistrovstvím světa, v každém případě o něco dříve než místní tradiční jazzový festival se odehrály v půli července v San Sebastianu jiné svátky, svátky středoevropského politického myšlení. Španělská sekce prestižní Asociace evropských novinářů pozvala již po šesté různorodé spektrum středoevropských politiků, publicistů a filosofů, aby početným posluchačům letního universitního semináře s názvem Střední Evropa mezi Bruselem a Moskvou přednesli své úvahy o tom, co se uprostřed kontinentu děje Dokonce i Australan se učil češtinu Praha - Na dvě stovky filologů, bohemistů a studentů češtiny z 28 zemí pěti kontinentů se včera v aule pražského Karolina rozloučily se svým měsíčním působením na Letní škole slovanských studií v Praze. Všem bylo předáno osvědčení o úspěšném absolvování tohoto kurzu. Pro zahraniční zájemce připravil prázdninové studium češtiny letos již po osmatřicáté Ústav bohemistických studií při Filozofické fakultě Univerzity Karlovy. Stejně jako v minulých letech byl o pobyt v Praze velký zájem i letos. Lidovci se

The Process of Morphological Annotation

- From tokenized to annotated text:


tokenized text (auto, w-layer)
(Auto) morphological analysis
morphological dictionary
text w/morph. interpretations
Manual morphological disambiguation (DA)
annotation guidelines
text w/select. interpretation
Manual adjudication
annotated text (m-layer)

Using the Results: Morphological Disambiguation

- Full morphological disambiguation
  - more complex than (e.g. English) POS tagging
- Three taggers:
  - (Pure) HMM
  - Feature-based (MaxEnt-like)
    - used in the PDT distribution
  - Voted Perceptron, (M. Collins, EMNLP'02)
- All: ~ 94-5% accuracy (perceptron is best)
  - rule & statistic combination: tiny improvement (Hajič et al., ACL 2001)

The Segmentation Problem: Possible solution (Arabic)

- Tokenization / segmentation not always trivial
  - Arabic, German, Chinese, Japanese
- Find max. no. of segments
  - 4 for Arabic
- expand every solution (morph. analysis) to the same number of segments, adding “blank” segments to the end
- concatenate tags (→ same length)
- concatenate “lemmas” (roots, ...)
- Result:
  - the same formal definition; can be converted back to segments trivially
  - tagging solves segmentation!

For your notes...

Part 2

DATA (continued): The Prague Dependency Treebank

The Prague Dependency Treebank and Valency Annotation (part 2)

Jan Hajič
Institute of Formal and Applied Linguistics
School of Computer Science
Faculty of Mathematics and Physics
Charles University, Prague
Czech Republic

PDT – Syntactic Annotation (tutorial part 2)

- Surface syntax annotation
  - Dependency surface syntax
  - Comparable to Penn Treebank annotation
    - Convertible: dependency ↔ parse trees
- Deep syntactic/semantic annotation
  - Dependency trees
  - Different topology
  - High level of generalization and formalization
  - Many node attributes

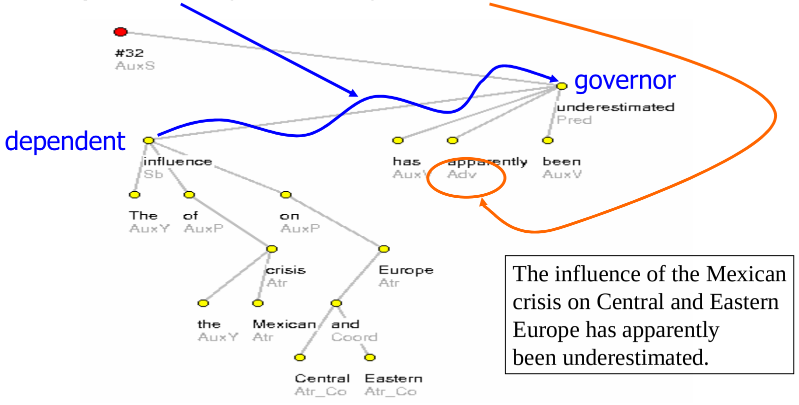
PDT Annotation Layers

- L0 (w) Words (tokens)
  - automatic segmentation and markup only
- L1 (m) Morphology
  - Tag (full morphology, 13 categories), lemma
- L2 (a) Analytical layer (surface syntax)
  - Dependency, analytical dependency function
- L3 (t) Tectogrammatical layer (“deep” syntax)
  - Dependency, functor (detailed), grammatemes, ellipsis solution, coreference, topic/focus (deep word order), valency lexicon

---

Layer 2 (a-layer): Analytical Syntax

- Dependency + Analytical Function



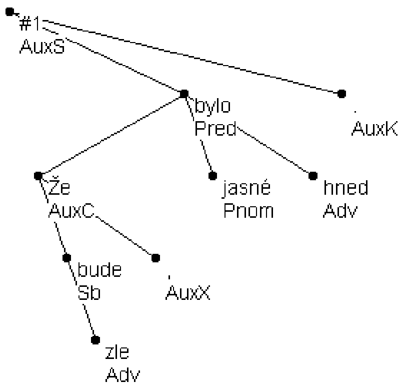
The influence of the Mexican crisis on Central and Eastern Europe has apparently been underestimated.

Analytical Syntax: Functions

- Main (for [main] semantic lexemes):
  - Pred, Sb, Obj, Adv, Atr, Atv(V), AuxV, Pnom
  - “Double” dependency: AtrAdv, AtrObj, AtrAtr
- Special (function words, punctuation,...):
  - Reflefives, particles: AuxT, AuxR, AuxO, AuxZ, AuxY
  - Prepositions/Conjunctions: AuxP, AuxC
  - Punctuation, Graphics: AuxX, AuxS, AuxG, AuxK
- Structural
  - Elipsis: ExD, Coordination etc.: Coord, Apos

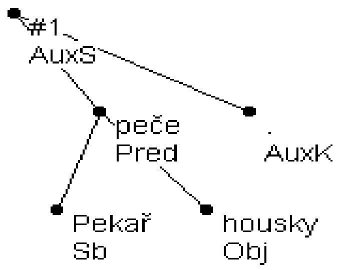
Example

- *lit.* That it will go wrong, (that) was clear immediately.
  - Že bude zle, bylo jasné hned.



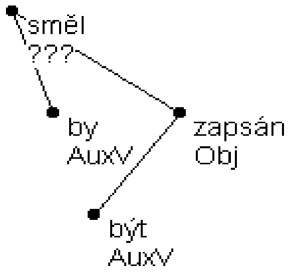
Surface Syntax Example

- Complete sentence: Sb, Pred, Obj
  - The-baker bakes rolls.
  - Pekař peče housky.



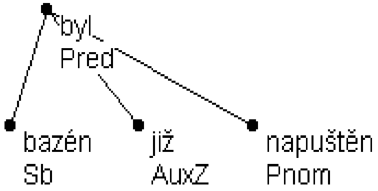
Surface Syntax Example

- Analytical verb form:
  - (he) allowed would-be to-be enrolled
  - směl by být zapsán



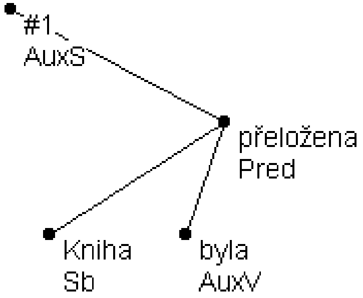
Surface Syntax Example

- Predicate with copula (state)
  - (the) pool has-been already filled
  - bazén byl již napuštěn



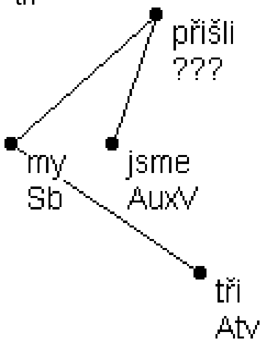
Surface Syntax Example

- Passive construction (action)
  - (The) book has-been translated [by Mr. X]
  - Kniha byla přeložena



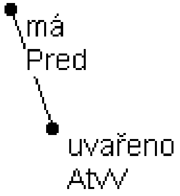
Surface Syntax Example

- Complement
  - we (are) came three
  - my jsme přišli tři



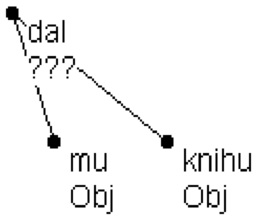
Surface Syntax Example

- Complement when NP is missing
  - (he) has cooked [his meals]
  - má uvařeno



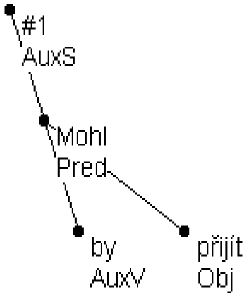
Surface Syntax Example

- Object
  - (he) gave him a-book
  - dal mu knihu



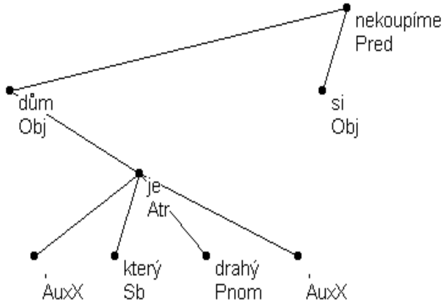
Surface Syntax Example

- Object used for infinitive of analytical verb forms
  - (he) Could come
  - Mohl by přijít



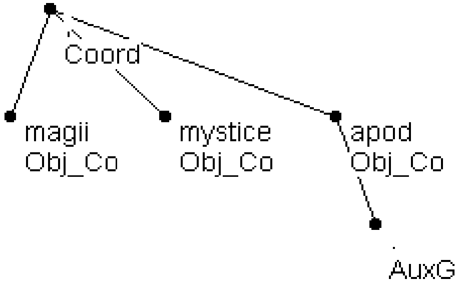
Surface Syntax Example

- Relative clause (embedded)
  - (a) house, which is expensive, (we) (to-ourselves) will-not-buy
  - dům , který je drahý , si nekoupíme



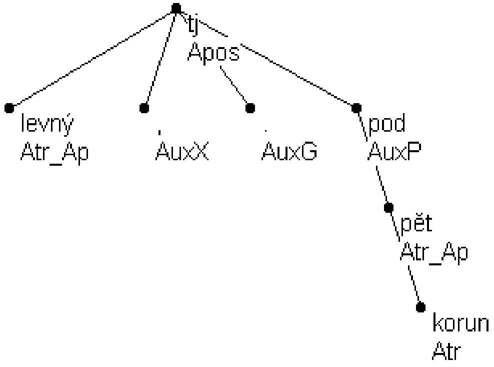
Surface Syntax Example

- Coordination
  - ... (to) magic, mystic(,) etc.
  - ... magii , mystice apod.



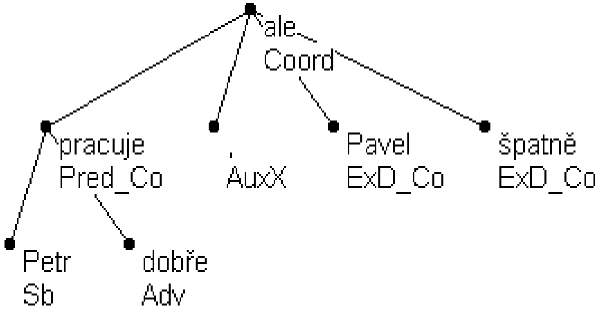
Surface Syntax Example

- Apposition
  - cheap, i.e. under 5 crown
  - levný , tj. pod 5 korun



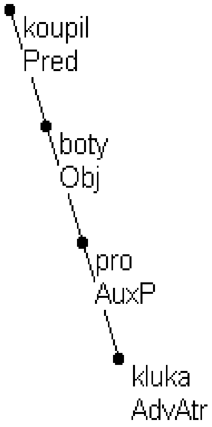
Surface Syntax Example

- Incomplete phrases
  - Peter works well , but Paul badly
  - Petr pracuje dobře, ale Pavel špatně



Surface Syntax Example

- Variants (equality)
  - (he) bought shoes for boy
  - koupil boty pro kluka



Using the Results: Parsing

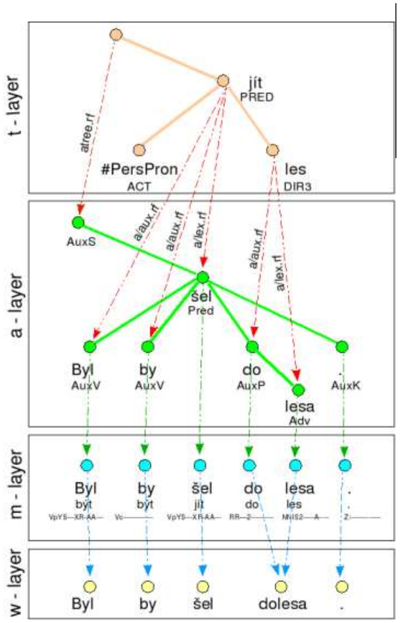
- Several parsers of Czech
  - Analytical layer dependency syntax
  - Trained on PDT 1.0 dat, 1.2 mil. words
- Collins (98), Charniak (00), Žabokrtský (02), Ribarov (04), Nivre (05), Zeman(05), McDonald (05)
- Best results (accuracy: percent of correct dependencies):
  - 84-85% for a single parser, > 86% for a combination

The Prague Markup Language (Intro only – see P. Pajas, p. 6)

- XML-based, UTF-8 coding used
- Stand-off annotation
  - strict hierarchical scheme
  - 4 files for each annotated document ~ 4 layers of annotation
- Can capture intermediate annotation
  - e.g., ambiguous analysis after morphological preprocessing
- Lexical resources linked in
  - valency lexicon referenced from t-layer data

XML Annotation Layers

- Strictly top-down links
- w+m+a can be easily "knitted"
- API for cross-layer access (programming)
- PML Schema / Relax NG
- [With slight modification, can be used for spoken data (audio as layer "-1")]



The Prague Markup Language Example

- m-layer data, linked to w-layer:

```
<m id="m-tr/_12941_01_00013.fs-s1w4">
  <src.rf>manual</src.rf>
  <w>
    <dest.rf>w#w-tr/_12941_01_00013.fs-s1w4</dest.rf>
    <trans>basic</trans>
  </w>
  <form>pocházela</form>
  <lemma>pocházet_:T</lemma>
  <tag>VpQW---XR-AA---</tag>
</m>
<m id="m-tr/_12941_01_00013.fs-s1w5">
  ...
```

Pointer to w-layer

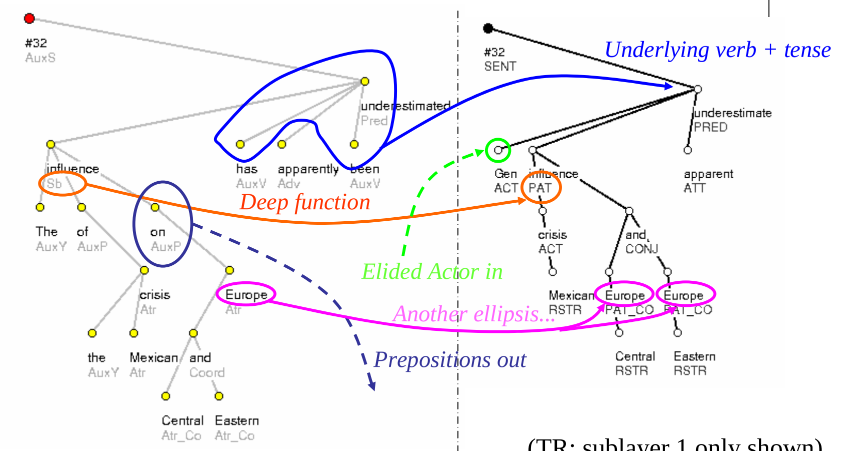
PDT Annotation Layers

- L0 (w) Words (tokens)
  - automatic segmentation and markup only
- L1 (m) Morphology
  - Tag (full morphology, 13 categories), lemma
- L2 (a) Analytical layer (surface syntax)
  - Dependency, analytical dependency function
- L3 (t) Tectogrammatical layer ("deep" syntax)
  - Dependency, functor (detailed), grammatemes, ellipsis solution, coreference, topic/focus (deep word order), valency lexicon

Layer 3 (t-layer): Tectogrammatical Annotation

- Underlying (deep) syntax
- 4 sublayers (integrated):
  - dependency structure, (detailed) functors
    - valency annotation
  - topic/focus and deep word order
  - coreference (mostly grammatical only)
  - all the rest (grammatemes):
    - detailed functors
    - underlying gender, number, ...
- Total
  - 39 attributes (vs. 5 at m-layer, 2 at a-layer)

Analytical vs. Tectogrammatical
annotation (TR: sublayer 1 only)



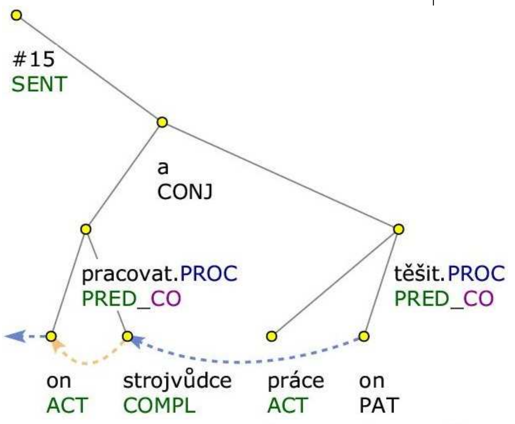
(TR: sublayer 1 only shown)

Layer 3: Tectogrammatical

- Underlying (deep) syntax
- 4 sublayers:
  - dependency structure, (detailed) functors
  - topic/focus and deep word order
  - coreference (mostly grammatical only)
  - all the rest (grammatemes):
    - detailed functors
    - underlying gender, number, ...

Example - TR

- Graphical visualization
- *He worked as an engineer and he liked the work.*



#15 Pracoval jako strojvůdce a práce ho těšila.

[He]worked as an-engineer and the-work him pleased.

Dependency Structure

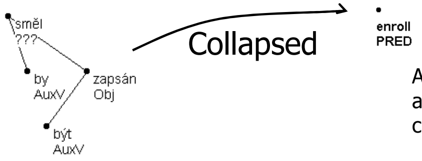
- Similar to the surface (Analytical) layer... ...but:
  - certain nodes deleted
    - auxiliaries, non-autosemantic words, punctuation
  - some nodes added
    - based on word (mostly verb, noun) valency
    - some ellipsis resolution
  - detailed dependency relation labels (functors)

Tectogrammatical Functors

- “Actants”: $\overbrace{\text{ACT, PAT,}}^{\text{syntactic}}\ \overbrace{\text{EFF, ADDR, ORIG}}^{\text{semantic}}$
  - modify: verbs, nouns, adjectives
  - cannot repeat in a clause, usually obligatory
- Free modifications (~ 50), semantically defined
  - can repeat; optional, sometimes obligatory
  - Ex.: LOC, DIR1, ...; TWHEN, TTILL,...; RSTR; BEN, ATT, ACMP, INTT, MANN; MAT, APP; ID, DPHR, ...
- Special
  - Coordination, Rhematizers, Foreign phrases,...

Tectogrammatical Example

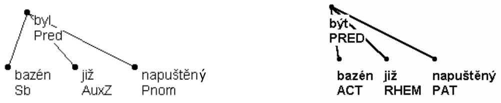
- Analytical verb form:
  - (he) allowed would-be to-be enrolled
  - směl by být zapsán



Additional attributes (grammatemes): conditional + “allow”

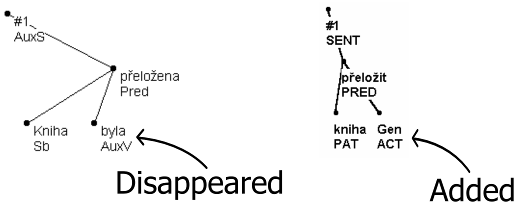
Tectogrammatical Example

- Predicate with copula (state)
  - (the) pool has-been already filled
  - bazén byl již napuštěný



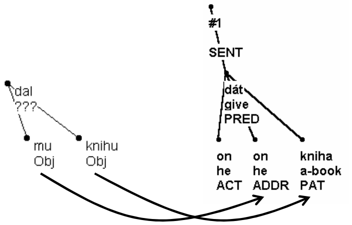
Tectogrammatical Example

- Passive construction (action)
  - (The) book has-been translated [by Mr. X]
  - Kniha byla přeložena



Tectogrammatical Example

- Object
  - (he) gave him a-book
  - dal mu knihu

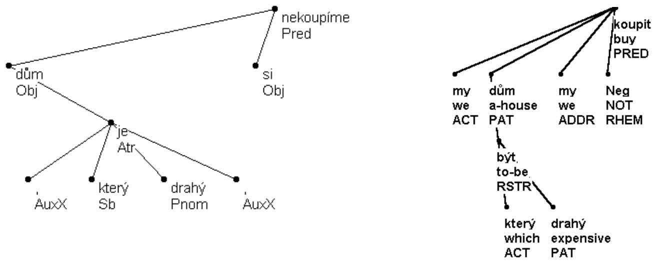


Obj goes into ACT, PAT, ADDR, EFF or ORIG based on governor’s valency frame

---

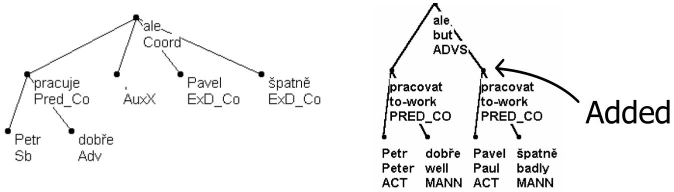
Tectogrammatical Example

- Relative clause (embedded)
  - (a) house, which is expensive, (we) (to-ourselves) will-not-buy
  - dům , který je drahý , si nekoupíme



Tectogrammatical Example

- Incomplete phrases
  - Peter works well , but Paul badly
  - Petr pracuje dobře, ale Pavel špatně

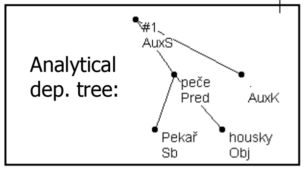


Layer 3: Tectogrammatical

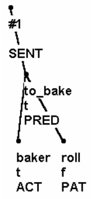
- Underlying (deep) syntax
- 4 sublayers:
  - dependency structure, (detailed) functors
  - topic/focus and deep word order
  - coreference (mostly grammatical only)
  - all the rest (grammatemes):
    - detailed functors
    - underlying gender, number, ...

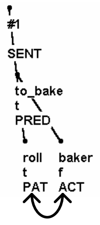
Deep Word Order, Topic/Focus (intro only: see E. Hajičová, p.3)

- Example:



  - Baker bakes rolls. vs. *Baker*^{IC} bakes rolls.





Deep Word Order Topic/Focus

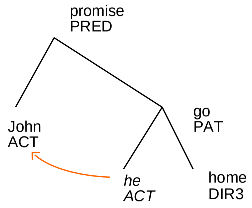
- Deep word order:
  - from “old” information to the “new” one (left-to-right) at every level (head included)
  - projectivity by definition (almost...)
    - i.e., partial level-based order -> total d.w.o.
- Topic/focus/contrastive topic
  - attribute of every node (t, f, c)
  - restricted by d.w.o. and other constraints

Layer 3: Tectogrammatical

- Underlying (deep) syntax
- 4 sublayers:
  - dependency structure, (detailed) functors
  - topic/focus and deep word order
  - coreference (mostly grammatical only)
  - all the rest (grammatemes):
    - detailed functors
    - underlying gender, number, ...

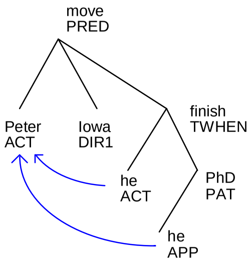
Coreference
(intro only: see E. Hajičová p.3)

- Grammatical (easy)
  - relative clauses
    - which, who
      - Peter and Paul, who ...
  - control
    - infinitival constructions
      - John promised to go ...
  - reflexive pronouns
    - {him,her,thme}self(-ves)
      - Mary saw herself in ...



Coreference

- Textual
  - Ex.: Peter moved to Iowa after he finished his PhD.



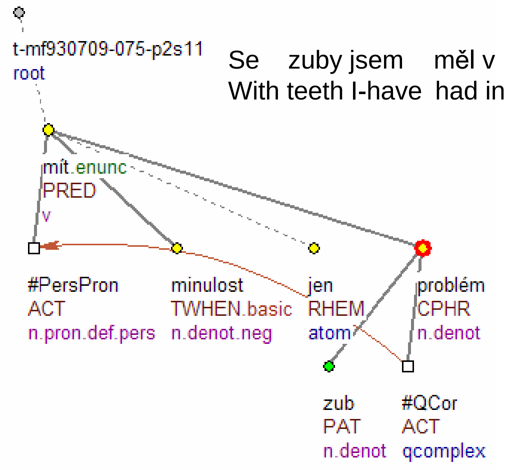
Layer 3: Tectogrammatical

- Underlying (deep) syntax
- 4 sublayers:
  - dependency structure, (detailed) functors
  - topic/focus and deep word order
  - coreference (mostly grammatical only)
  - all the rest (grammatemes):
    - detailed functors
    - underlying gender, number, ...

Grammatemes
(intro only: see Z. Žabokrtský p. 3)

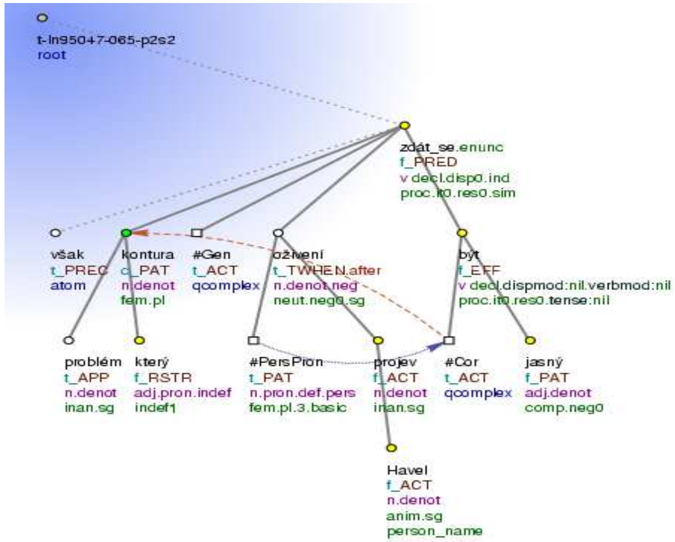
- Detailed functors (subfunctors)
  - only for some functors:
    - TWHEN: before/after
    - LOC: next-to, behind, in-front-of, ...
    - also: ACMP, BEN, CPR, DIR1, DIR2, DIR3, EXT
- Lexical (underlying)
  - number (SG/PL), tense, modality, degree of comparison, ...
  - strictly only where necessary (agreement!)

Example - simplified view



Se zuby jsem měl v minulosti jen problémy.
With teeth I-have had in the-past only problems.

Fully Annotated Sentence



The boundaries of some problems seem to be clearer after they were revived by Havel's speech.

Part 3

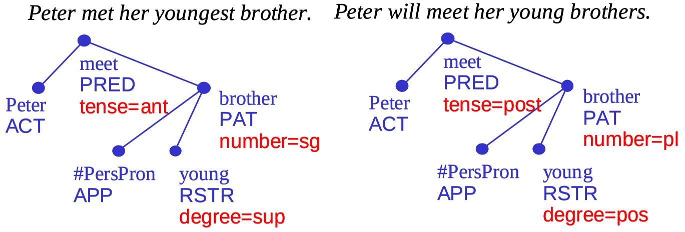
DATA (continued): The Prague Dependency Treebank

Grammatemes in the PDT 2.0

Zdeněk Žabokrtský
Dept. of Formal and Applied Linguistics
Charles University, Prague
zabokrtsky@ufal.mff.cuni.cz

PDT 2.0

What is a "grammateme"?



- the same t-lemmas, the same tree topology, the same functors, but the original sentences are obviously not synonymous and must be distinguished at the t-layer (must obtain different t-trees) !
- the difference is in grammatemes ~ t-node attribute-value pairs representing morphological meanings (semantically indispensable morphological categories)
- e.g. number for nouns, tense for verbs, degree for adjectives, deontic/verb/sentence modality ...

PDT 2.0

What is not a grammateme?

- grammatemes are not just straightforward counterparts of surface morphological categories (as stored in m-layer tags) !
- some morphological categories are only imposed by grammar and thus are not semantically relevant
  - gender, number or case of an adjective in a noun group come from agreement with the noun (e.g. in Czech or German), not from semantics
  - similarly, person is not a grammateme of verbs, as it is only induced by subject-verb agreement
- on the surface, grammatemes can be expressed both inflectionally and analytically -> info about grammatemes can be distributed over more than one m-layer token
  - comparative of adjectives in English (*more interesting*)
  - future tense of imperfectives in Czech (*budu chodit.../ I will go...*)

PDT 2.0

Complete list of grammateme attributes used in PDT 2.0

1. **gram/number** - number of semantic nouns
2. **gram/gender** - gender of semantic nouns
3. **gram/person** - person of pronominal semantic nouns
4. **gram/politeness** -basic vs. polite/esteemed form, relevant for pronominal semantic nouns
5. **gram/indeftype** (type of indefiniteness of pro-forms)
6. **gram/numertype** (type of numeric expression)
7. **gram/negation** - negation of semantic nouns, adjectives, and adverbs (not of verbs)
8. **gram/degcmp** - degree of comparison of semantic adjectives and adverbs
9. **gram/tense** - tense of verbs
10. **gram/aspect** - aspect of verbs
11. **gram/verbmod** - basic verb modality (indicative, imperative, conditional)
12. **gram/deontmod** - deontic modality expressed by modal verbs
13. **gram/dispmod** - dispositional modality (specific for Czech)
14. **gram/resultative** - resultativeness of verbs
15. **gram/iterativeness** - iterativeness of verbs
16. **sentmod** - sentence modality (enunciative, exclamative, desiderative, imperative, interrogative)

PDT 2.0

Grammateme number

- values:
  - sg - singular
  - pl - plural
  - nr - not recognized

- m-layer/t-layer asymmetry:
  - pluralia tantum: *jedny dveře/ dvoje dveře* (one door, two doors) - only the plural form exists at the m-layer, but sg/pl should be disambiguated at the t-layer
  - polite form: *"Viděl jste to, Petře?"* (Did you see it, Petr?) - complex verb form containing an auxiliary verb in plural at the m-layer, but at the t-layer the grammateme number (filled in the reconstructed #PersPron node) is equal to singular

---

PDT 2.0

Grammateme tense

- relative tense of verbs (with respect to the tense of the governing clause)

- values:
  - sim - simultaneous
  - ant - anterior
  - post - posterior
  - nil - absent (with infinitives)
  - nr - not recognized

- m-layer means for expressing tense=post in Czech:
  - inflection with perfectives (*uvařím* - I will cook)
  - auxiliary verb *být* with imperfectives (*budu zpívat* - I will sing)
  - prefix *po-/ pů-* with a limited set of verbs (*pojedu* - I will go)

PDT 2.0

Grammateme indeftype (I)

- pro-form - a word used to replace or substitute other words, phrases, clauses...
- pronouns (pro-nouns), pro-adjectives, pro-numerals, pro-adverbs
- there are many semantically significant analogies present in the pro-forms systems, but usually not explicitly distinguished in the POS tag sets
- example of such parallelism:
  - nobody/never/nowhere... vs. everybody/always/everywhere...
- grammateme indeftype (type of indefiniteness) dedicated for all indefinite pro-forms
- to capture the parallelisms, each group of pro-forms is represented with t_lemma identical with the relative form: *někde->kde (nowhere->where), kdokoli->kdo (whoever->who), nikdy->kdy (never->when)*

PDT 2.0

Grammateme indeftype (II)

t-lemma:	***kdo***	***co***	***který***	***jaký***
value of the grammateme **indeftype**: relat	*kdo*	*co*	*který, jenž*	*jaký*
indef1	*někdo*	*něco*	*některý*	*nějaký*
indef2	*kdosi, kdos*	*cosi, cos*	*kterýsi*	*jakýsi*
indef3	*kdokoli(v)*	*cokoli(v)…*	*kterýkoli(v)*	*jakýkoli(v)*
indef4	*ledakdo, leckdo…*	*ledaco, lecco…*	*leckterý, ledakterý*	*lecjaký, ledajaký*
indef5	*kdekdo*	*kdeco*	*kdekterý*	*kdejaký*
indef6	*málokdo, kdovíkdo…*	*máloco…*	*málokterý…*	*všelijaký…*
inter	*kdo, kdopak…*	*co, copak…*	*který, kterýpak*	*jaký, jakýpak*
negat	*nikdo*	*nic*	*žádný*	*nijaký*
total1	*všechen*	*všechen, všechno, vše*	–	–
total2	–	–	*každý*	–

Grammateme indeftype (III)

- indefinite, negative, interrogative, and relative pronouns and other pro-forms are unproductive classes with (at least to a certain extent) transparent derivational relations also in other languages
- preliminary sketch of several English and German pronouns classified by indeftype

	English	English	German	German
Lemma	*who*	*what*	*wer*	*was*
indeftype:				
relat	who	what	wer	was
indef1	someone	something	jemand	etwas
indef2	-	-	irgendjemand	irgendetwas
indef3	whoever	whatever	-	-
inter	who	what	wer	was
negat	nobody	nothing	niemand	nichts
total1	all	everything	alle	alles
total2	each	each	jeder	jedes

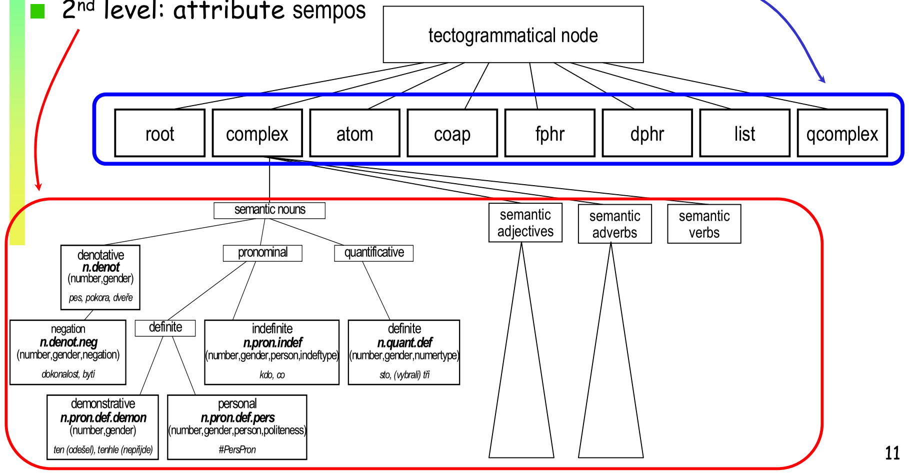
Typing of t-nodes

- unlike t_lemmas and functors, grammateme attributes are not relevant for all t-nodes
  - obviously, no tense for *dog*, no degree of comparison for *(he) waits*, etc.
- crucial question: how to formally declare presence/absence of a certain grammateme in a certain t-node ? → the need for node typing
- our solution: two-level hierarchy of node types
  - 1st level: 8 coarse-grained types of nodes
  - 2nd level: 19 more specific subtypes, corresponding to detailed semantic parts of speech

---

Two-level hierarchy of t-node types

- 1st level: attribute nodetype
- 2nd level: attribute sempos



tectogrammatical node

root
complex
atom
coap
fphr
dphr
list
qcomplex

semantic nouns
semantic adjectives
semantic adverbs
semantic verbs

denotative
n.denot
(number,gender)
pes, pokora, dveře

pronominal

quantificative

negation
n.denot.neg
(number,gender,negation)
dokonalost, bytí

definite

indefinite
n.pron.indef
(number,gender,person,indeftype)
kdo, co

definite
n.quant.def
(number,gender,numertype)
sto, (vybrali) tři

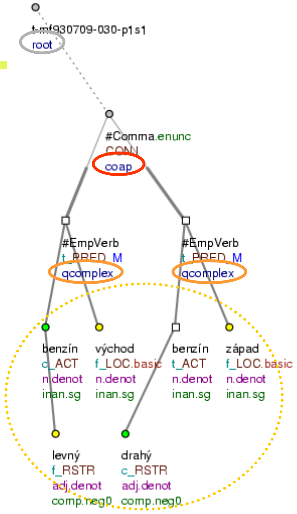
demonstrative
n.pron.def.demon
(number,gender)
ten (odešel), tenhle (nepřijde)

personal
n.pron.def.pers
(number,gender,person,politeness)
#PersPron

First level of the hierarchy: attribute nodetype

- 8 nodetype values:

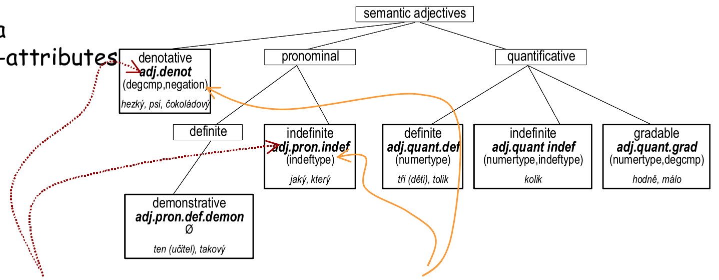
  root | complex | qcomplex | list | atom | coap | dphr | fphr
- fully automatic annotation - use of
  - the tree structure → root
  - t-attributes
    - t-lemma → qcomplex | list
    - functor → atom | coap | dphr | fphr
  - otherwise → complex



Levnější benzín na Východě, dražší na Západě
Cheaper gasoline in the East, more expensive one in the West

Second level of the hierarchy: attribute sempos

- sempos relevant only for nodetype=complex t-nodes
- 19 values of the attribute sempos:
  - n. ... | adj. ... | adv. ... | v. ...
- fully automatic annotation – use of
  - m-tag
  - t-lemma
  - other t-attributes


semantic adjectives

denotative
adj.denot
(degcmp,negation)
hezký, psí, čokoládový

pronominal

definite

demonstrative
adj.pron.def.demon
Ø
ten (učitel), takový

indefinite
adj.pron.indef
(indeftype)
jaký, který

quantificative

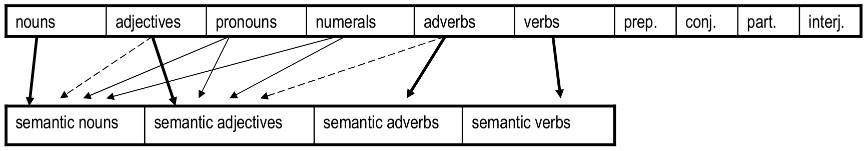
definite
adj.quant.def
(numertype)
tři (děti), tolik

indefinite
adj.quant indef
(numertype,indeftype)
kolik

gradable
adj.quant.grad
(numertype,degcmp)
hodně, málo

- sempos value delimits the set of relevant grammatemes

M-layer POS tags vs. sempos


nouns	adjectives	pronouns	numerals	adverbs	verbs	prep.	conj.	part.	interj.

semantic nouns	semantic adjectives	semantic adverbs	semantic verbs

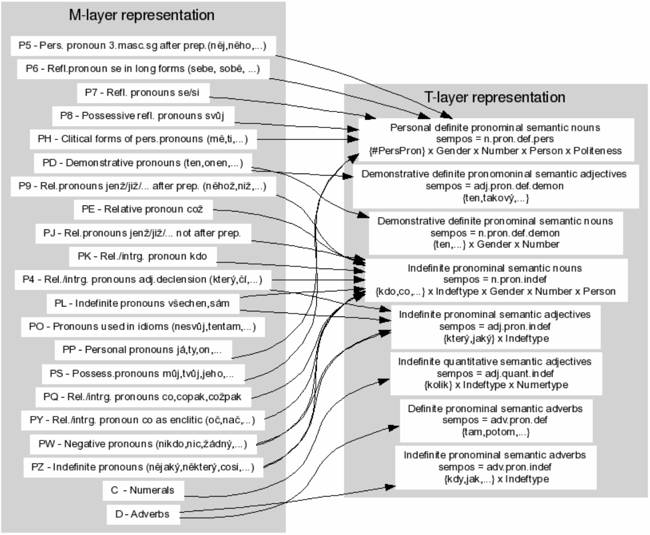
⟵ "prototypical" relations between semantic and "traditional" parts of speech

← distribution of pronouns and numerals into semantic parts of speech

⇠ classification following the derivational information

- Examples of asymmetry:
  - m-layer possessive adjectives (e.g. *matčin*/mother's) converted to semantic nouns (*matka*/mother)
  - m-layer deadjectival adverbs (*pěkně*/nicely) converted to semantic adjectives (*pěkný*/nice)

Pro-forms: m-layer tags vs. t-layer sempos


M-layer representation

P5 - Pers. pronoun 3.masc.sg after prep.(něj,něho,...)
P6 - Refl.pronoun se in long forms (sebe, sobě, ...)
P7 - Refl. pronouns se/si
P8 - Possessive refl. pronouns svůj
PH - Clitical forms of pers.pronouns (mě,ti,...)
PD - Demonstrative pronouns (ten,onen,...)
P9 - Rel.pronouns jenž/již/... after prep. (něhož,niž,...)
PE - Relative pronoun což
PJ - Rel.pronouns jenž/již/... not after prep.
PK - Rel./intrg. pronoun kdo
P4 - Rel./intrg. pronouns adj.declension (který,čí,...)
PL - Indefinite pronouns všechen,sám
PO - Pronouns used in idioms (nesvůj,tentam,...)
PP - Personal pronouns já,ty,on,...
PS - Possess.pronouns můj,tvůj,jeho,...
PQ - Rel./intrg. pronouns co,copak,cožpak
PY - Rel./intrg. pronoun co as enclitic (oč,nač,...)
PW - Negative pronouns (nikdo,nic,žádný,...)
PZ - Indefinite pronouns (nějaký,některý,cosi,...)
C - Numerals
D - Adverbs

T-layer representation

Personal definite pronominal semantic nouns
sempos = n.pron.def.pers
{#PersPron} x Gender x Number x Person x Politeness

Demonstrative definite pronomoninal semantic adjectives
sempos = adj.pron.def.demon
{ten,takový,...}

Demonstrative definite pronominal semantic nouns
sempos = n.pron.def.demon
{ten,...} x Gender x Number

Indefinite pronominal semantic nouns
sempos = n.pron.indef
{kdo,co,...} x Indeftype x Gender x Number x Person

Indefinite pronominal semantic adjectives
sempos = adj.pron.indef
{který,jaký} x Indeftype

Indefinite quantitative semantic adjectives
sempos = adj.quant.indef
{kolik} x Indeftype x Numertype

Definite pronominal semantic adverbs
sempos = adv.pron.def
{tam,potom,...}

Indefinite pronominal semantic adverbs
sempos = adv.pron.indef
{kdy,jak,...} x Indeftype

Grammatemes: Annotation process

- implementation: 2000 Perl LOCs in the ntred environment
- 2000 lines of linguistic rules in a special notation
- extensive usage of m-layer and a-layer manual annotation -> mostly automatic annotation possible
- only 5 man-months of human annotation

PDT 2.0

More reading about grammatemes

- Chapter 2.4 in the t-layer manual (included in the PDT 2.0 documentation)
- Razímová, M., Žabokrtský, Z.: *Morphological Meanings in the Prague Dependency Treebank 2.0.* In: Proceedings of TSD. 2005
- Razímová, M., Žabokrtský, Z.: *Annotation of Grammatemes in the Prague Dependency Treebank 2.0.* Proceedings of Annotation Science Workshop, LREC. 2006
- Ševčíková Razímová, M., Žabokrtský, Z.: *Systematic Parametrized Description of Pro-forms in the Prague Dependency Treebank 2.0.* In: Proceedings of TLT. 2006

Annotation of the Topic-Focus Articulation in the Prague Dependency Treebank

Eva Hajičová

Overview

1. Linguistic motivation of TFA annotation:
    i. Basic notions
    ii. Why TFA should be annotated in the TGTS's: semantic relevance of TFA
2. TGTS attribute TFA and its values
3. Examples
4. Testing linguistic hypotheses on a deep layer of corpus annotation

---

Basic notions of TFA

- Information structure of the sentence
    - ○ Topic-focus articulation
        - Topic, theme, …
        - Focus, rheme, …
    - ○ based on *given* x *new*, but not identical to this cognitive dichotomy:
        - John and Mary entered the dining-room. They first went to the window …
        - Mary Called Jim a Republican. Then he insulted HER.
        - Mary called Jim a republican. Then he INSULTED her.

Semantic relevance of TFA

- Everybody in this room knows two languages.
  Two languages are known by everybody in this room.
- Many men read few books.
  Few books are read by many men
- Smoke in the hallway!
  In the hallway, you smoke.
- Staff behind the COUNTER.
  STAFF behind the counter.
- Carry DOGS.
  CARRY dogs. Dogs must be carried.

Topic-focus articulation in PDT

one attribute (TFA – topic-focus articulation) with values concerning the *contextual boundness* of the nodes

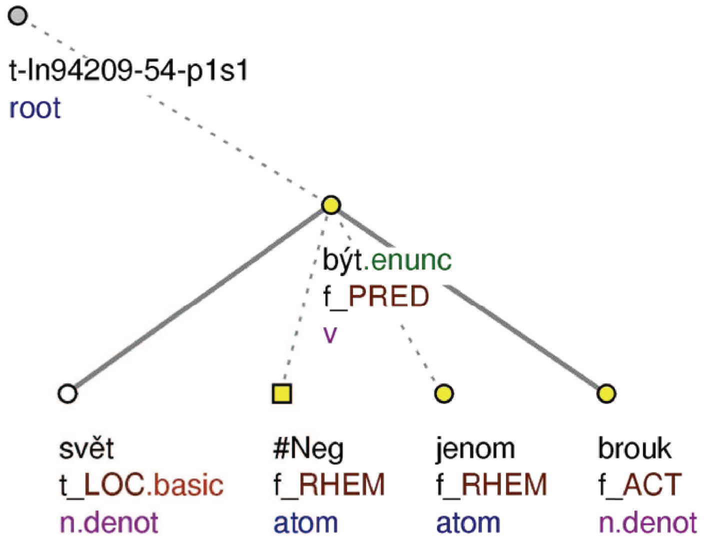
three values in the TFA attribute:

t – contextually bound non-contrastive

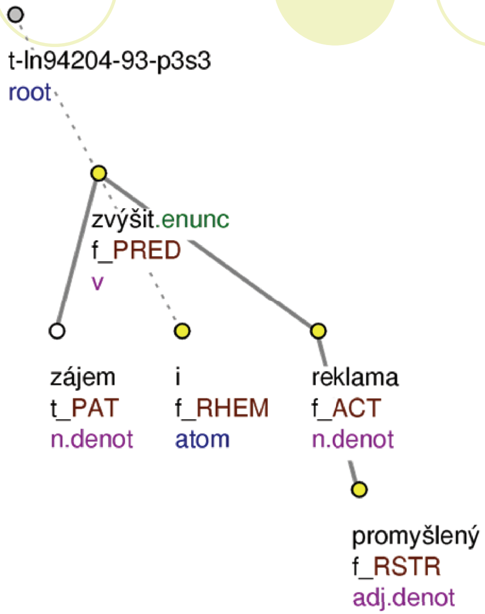
c – contextually bound contrastive

f – contextually non-bound

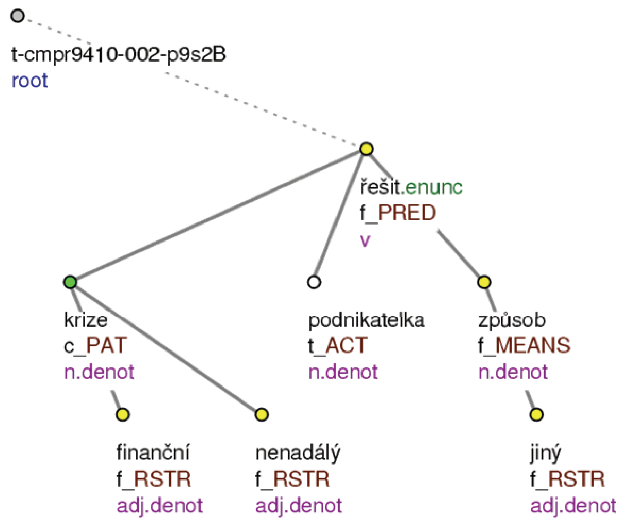
Example *Na světě nejsou jenom brouci. [In the-world there-are-not only beetles.]*



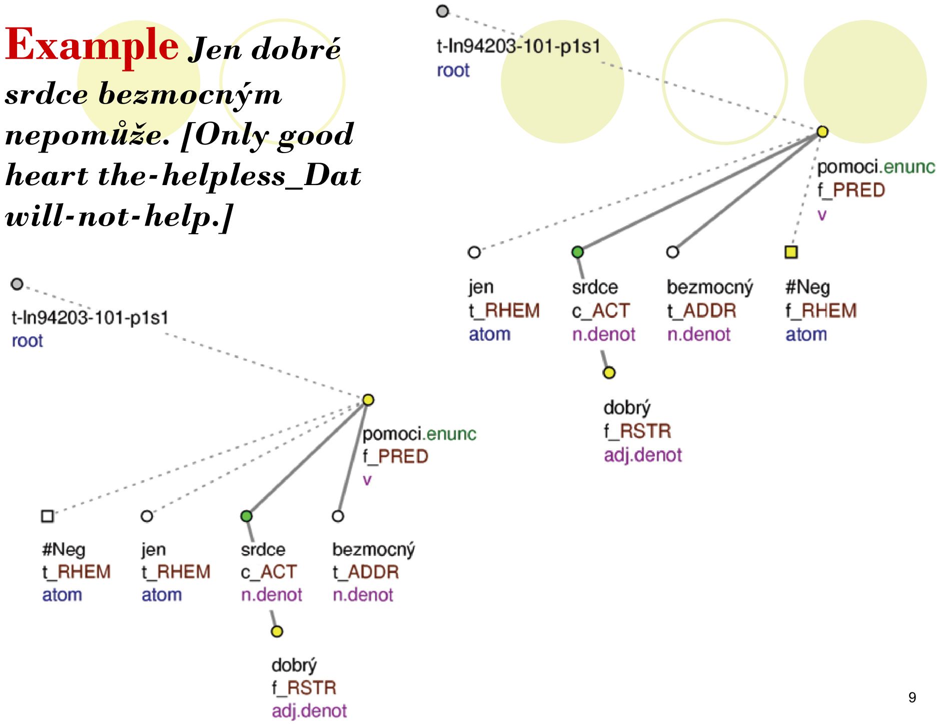
Example *Zájem zvýšila i promyšlená reklama. [The-interest_Acc raised also the-sophisticated campaign.]*



Example *Nenadálou finanční krizi podnikatelka řešila jiným způsobem. [The-sudden financial crisis_Acc the-entrepreneur_Nom solved by other means.]*



Example *Jen dobré srdce bezmocným nepomůže. [Only good heart the-helpless_Dat will-not-help.]*



Testing linguistic hypotheses

- Corpus annotation is not a self-contained task.
- A necessary condition for a usable annotated corpus: based on a sound linguistic theory.
- PDT: linguistic basis: Functional Generative Description.
- One of the important uses of corpus: test for linguistic theories.

Hypothesis A1: Division into T and F based on boundness

Hypothesis A1:

- **the global division of the sentence into its TOPIC (what the sentence is about) and its FOCUS (what is said about the topic) can be made on the basis of boundness**

Sgall (1979; see also Sgall et al. 1986 216f),

original algorithm implemented and tested on the whole of PDT; the results reported in Hajičová, Havelka and Veselá (2005)

Hypothesis A1 (cont.)

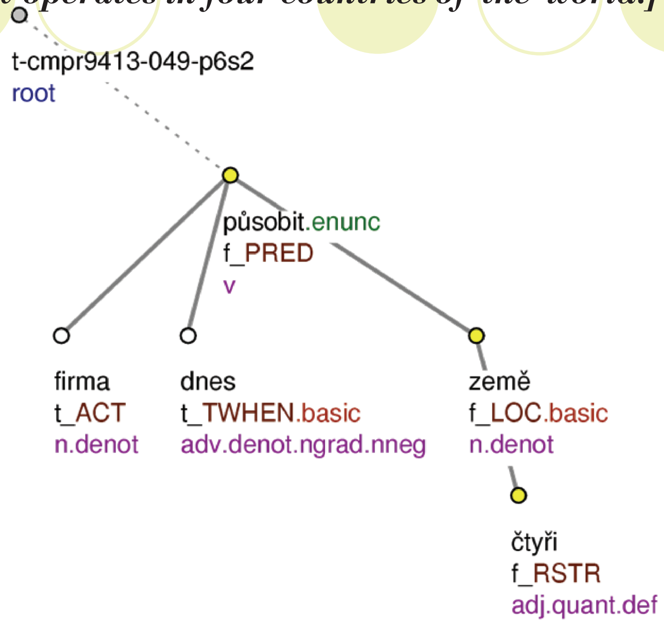
a) main verb=f → belongs Focus (F); else, → T

b) all the nodes directly dependent on the main verb and carrying t → T, together with all nodes depending on them

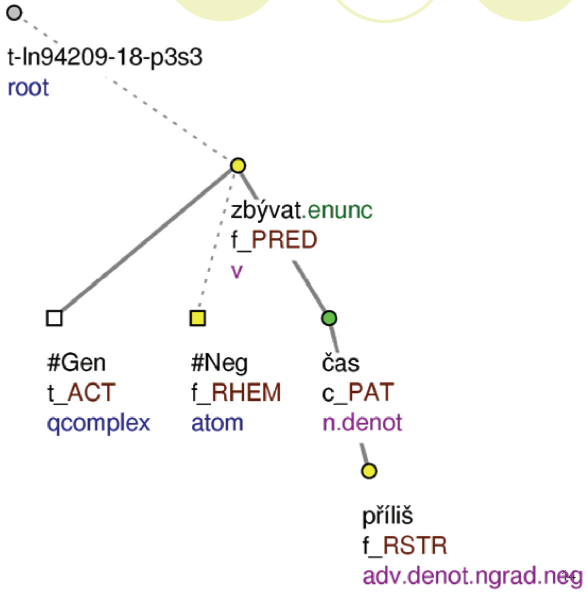
c) all the nodes directly dependent on the main verb and carrying f →F, together with all nodes depending on them

d) main verb = t & all nodes directly depending on the main verb = t: follow the rightmost edge leading from the main verb to the first node(s) on this path carrying the value f → this/these node(s) and all the nodes depending on it/them = F

Example *Firma dnes působí ve čtyřech zemích světa. [The-firm now operates in four countries of-the-world.]*



Example *Času příliš nezbývá. [Time_Gen too-much does-not-remain.]*



Results of the implementation

● F: V + subtrees	85,7%
● F: right-attached subtrees of V.t	8,58%
● Quasi-focus	4,41%
● F interrupted by c-node	0,06%
● Ambiguous partition	1,14%
● No focus indentified	0,11%

Results

- **in Czech**: the **boundary** between Topic and Focus can be determined in principle on the basis of the consideration of the **status of the main predicate** and its direct dependents.
- TFA annotation leads to satisfactory results in cases of rather **complicated "real" sentences** in the corpus.

Certain modification of the annotation procedure necessary, but the material gathered and analyzed in this way may be further used for the study of several aspects of the **discourse patterning.**

Hypothesis A2: The so-called systemic ordering

Hypothesis A2:

In the focus part of the sentence the complementations of the verb (be they arguments or adjuncts) follow a certain canonical order (not necessarily the same for all languages).

tested with a series of psycholinguistic experiments (with speakers of Czech, German and English) but PDT offers a richer and more consistent material → work in progress (Lešnerová)

Hypothesis A2: (cont.)

Tested on PDT

a) the **F** of the sentence identified (see A1)

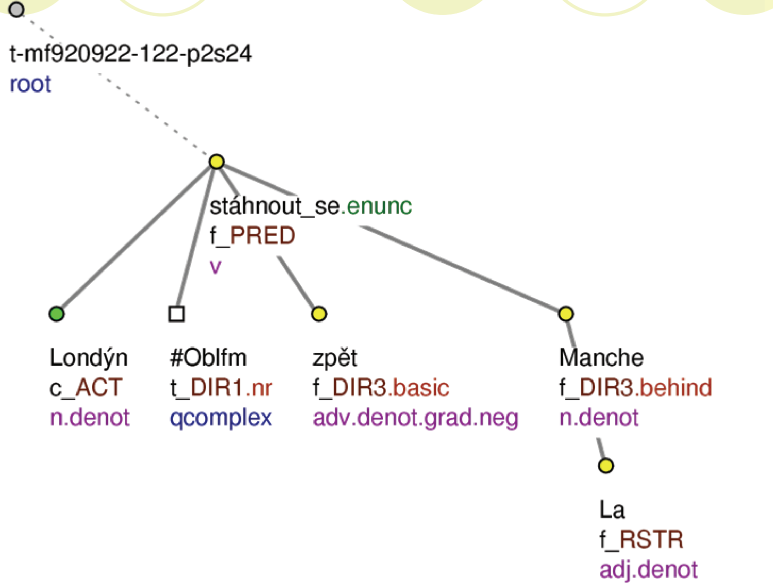
b) in TGTS: the **surface order in F** preserved

c) **systemic ordering** hypothetically stated

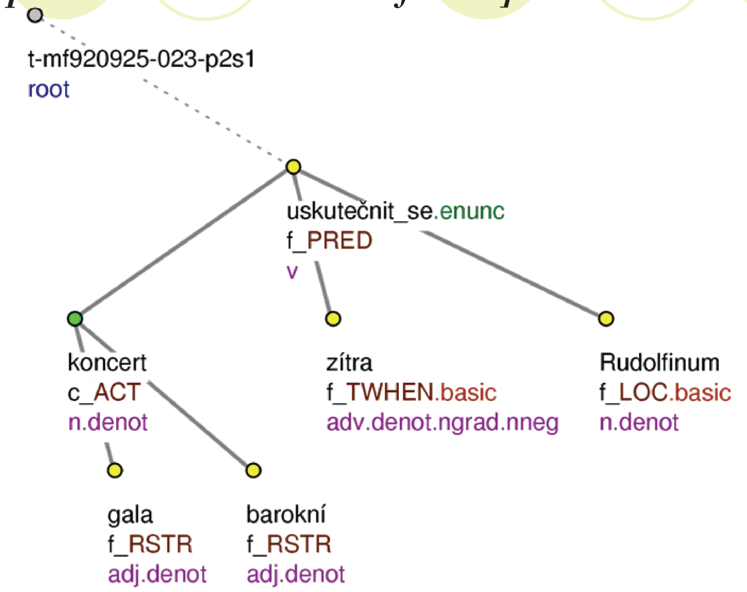
→these pieces of information used to compare the order of the complementations in the actual sentence and the assumed order according to the scale of systemic ordering

---

Example *Londýn se stáhl zpět za La Manche. [London Refl. Withdrew back behind La Manche.]*



Example *Barokní gala koncert se uskuteční zítra v Rudolfinu. [A-baroque gala concert Refl. Wiil-take-place tomorrow at Rudolfinum.]*



Conclusions

- → importance of the deep-layer corpus annotation for the **study** of most various language phenomena
- → the tectogrammatical layer of annotation brings about an indispensable source of information for **testing** any linguistic **theory** and any **grammar build-up**

Coreferential Relations in the Prague Dependency Treebank

Eva Hajičová

PDT layers and coreference

- The three PDT layers – capture grammatical information
- Coreference relations – textual relations – “beyond” grammar

BUT:

the aim: by annotating these relations to get more insight into the inter- and intrasentential structure

Tectogrammatical annotation

- semiautomatic → user-friendly tree editor (TRED)
- 3 steps (phases):
  - build-up of underlying syntactic tree structures (incl. nodes deleted on the shallow structure) and assigning the nodes functional labels
  - adding the values of the topic-focus attribute
  - adding the coreferential links

Annotation of coreference relations in PDT

- coreference relations in the narrower sense
- a binary relation between an anaphor and an antecedent:
  - the antecedent may be in a different TGTS
  - the antecedent may also be an entity that is not represented in any TGTS
- 2 kinds of coreference
  - grammatical
  - textual

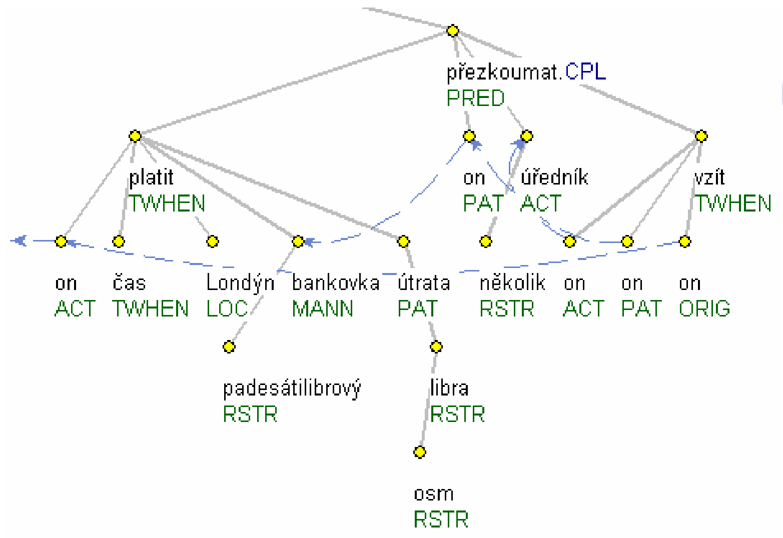
Annotational scheme

- explicit coreference links are technically represented as pointers (pml reference) leading form anaphor t-nodes to their antecedent t-nodes
- three coreferential attributes with an anaphor:
  - **coref_gram.rf** – identifier (or a list of identifiers) of the antecedent(s) in the sense of grammatical coreference
  - **coref_text.rf** – identifier (or a list of identifiers) of the antecedent(s) in the sense of textual coreference
  - **coref_special** – special types of coreference:
    - 1. **segm –** coreference with a sequence of preceding sentences (further underspecified)
    - 2. **exoph –** antecedent not present in the text at all

---

Notational devices for coreferential links in PDT

- arrows from the anaphor to the antecedent(s)
- different colours of the arrows according to the type of coreference
- special devices: an exophora, a segment
- an annotator-friendly special module within the TRED editor



Lit.: (*When*) *time-ago he paid in-London with-50-pound banknote expenditure of-8 pounds, checked it several clerks (before) they took it from-him.*

Grammatical coreference

- verbs (nouns, adjectives) of control
  - ○ *John asked Mary to **[0]** come.*
- reflexive pronouns
  - ○ *John shaved **himself**.*
- relative pronouns
  - ○ *John, **who** came late, apologized.*
- verbal complements
  - ○ *John came **[0]** bare-footed.*
- reciprocity
  - ○ *John and Mary kissed **[0]**.*

Textual coreference

- Present stage:
  - ○ in the whole PDT 2.0
    - demonstrative and anaphoric pronouns (also in their zero form), 3rd person
    - bridging anaphora is not included
  - ○ in a sample of 80 PDT documents
    - anaphoric relations leading from nouns incl. a rough classification of bridging anaphora

---

Types of textual coreference

- link to a particular node
- link to the governing node of a subtree
- segm(ent): referent is a whole segment of text
- exoph(or): referent is „out“ of co-text
- unsp(ecified): reference is difficult to be specified

Link to a particular node

- this node represents an antecedent of the anaphor:

*Do you think that the decision of NATO whether **[it]** will be enlarged or not will depend on the attitude of Russia?*

→ the link from ***it*** leads to ***NATO***

Link to the governing node of a subtree

- antecedent is represented by this node plus (some of) its dependents; also the way how a link to a previous/following clause or a whole previous sentence is being established:

*But it is a different thing when someone is an entrepreneur and then goes into politics than when political changes elevate somebody to the top and he then uses **this** in his economic activities.*

→ the link from ***this*** points to the root of the tree (*elevate*) = to the main verb of the second conjunct.

Segm(ent)

- referent is a whole segment of (previous) text larger than one sentence (phrase):

*According to Kohl it should not be forgotten that on June 22, 1941 Germany attacked the Soviet Union. Germans on behalf of Germany caused the Russians to suffer immensely. It also cannot be forgotten what the Russians did to Germans. From all **this** we should learn.*

Segm(ent) 2

- includes also the cases, when the antecedent is understood by inferencing from a broader co-text:

*The big shots buy in a bank for ten and sell for fifteen. But this leads to a rapid transformation. The acrages of about 25 ha disappear, the number of owners raises to 500. I guess that within two years they will be able to pay back the debt to the bank and in the third year they will work for themselves. And they will hire only capable people, it will be in their best interest. Those who understand **this**, will have an advantage.*

Exoph(or)

- a specifically marked link denoting that the referent is “out“ of the co-text, it is known only from the situation:

*In the height of summer 1939 only a few people could believe the hopeful words Chamberlain uttered […] after the return from Munich: I think that **this** is peace for our time.*

→ ***this*** = Munich Treaty

Unsp(ecified)

- a specific mark reserved for cases of reference difficult to be identified; a decision is not to be made between two or more referents but that the reference cannot be specified even if the situation is taken into account:

*The disappearance of the medical instrument weighing 700 kg **[they]** announced on June 30th this year. According to the information of LN, however, the radiator disappeared by the end of the last year.*

Statistics: volume of data

number of annotated documents (i.e. the whole PDT 2.0 t-layer data)	3 165
number of sentences/t-trees	49 431
number of t-nodes	724 396
total number of co-refering t-nodes	46 242 (6.3% of all)

Statistics: types of coreference

grammatical coreference	23 252 (50.3%)
textual coreference	22 368 (48.4%)
special types	
segm	505 (1.1%)
exoph	120 (0.2%)

Statistics: t-lemmas with anaphors (1)

most frequent t-lemmas with **grammatical** coreference	
1. který	7 435 (32% of all grammatical)
2. #Cor	5 907 (25%)
3. #PersPron	4 419 (19%)
4. #QCor	2 472 (10%)
5. #Rcp	1 114 (4.7%)
6. co	575 (2.5%)
7. kde	555 (2.3%)
…	

Statistics: t-lemmas with anaphors (2)

most frequent t-lemmas with **textual** coreference	
1. #PersPron	18 622 (83%)
2. ten	3 733 (16.7%)
…	

Statistics: expressed vs. restored

grammatical coreference	
anaphors expressed in the surface shape	13 783 (59.3%)
restore anaphor nodes	9 469 (40.7%)
textual coreference	
anaphors expressed in the surface shape	11 131 (49.7%)
restored anaphor nodes	11 237 (50.3%)

Steps beyond: segm(ent)

The boundaries of the (relevant) segment are not quite clear:

The only reason for me to stay in America is money. […] In America, I rent a house every year and at the end of the season I rush home. I have friends here, we go fishing, we play tennis, we visit each other. I often visit my parents in Martin. I am simply at home here. […] In Canada ***this*** *is totally different.*

Steps beyond: exoph(ora)

Border-line between exophora and other types of coreferential relations:

→ coreference to an unspecified element:

*A well-known native of Pardubice, Roman M. […] had drunk himself to death after he found out that he was born in Hradec Králové. […] The birth of children from Pardubice in Hradec Králové periodically happens. Once in every two years **[they]** brought them here, said the nurse at the obstetric clinic of the Hradec hospital.*

→ coreference to a segment („inferential“ type):

*Sad people write bright merry books and merry people write sad [ones]. One has to balance **it** somehow.*

Pronoun with other than referential function

- Intensifying function – particle *to* (*ten*):
  - *Boy, is it raining! Lit. **[that]** but it-rains! = meaning: it rains very much.*
- Conceptually „empty“ occurrences:
  - *As I have imagined for a long time her trip abroad, to Spain or Greece, where [lit.] **it** draws her.*
- Phrasemes
  - *Lit. **That** you-have hard, this young person's father has connections.*

Open questions (1)

Coreferential link leads to the root

× antecedent is a part of sentence:

*When Jiří Krupička sent me the manuscript of his Renaissance of Reason, which has been published now in the publishing house Český spisovatel, and I looked into it for the first time, not only my knees but also my heart trembled. And **this** [happened] for several reasons.*

Open questions (2)

With a coreferential chain, all links are established:

*The agreement of course has not solved anything – it only deepened the feeling in the **protestants** that London leaves **them** in the lurch. Today this feeling, that **[they]** are only a burden for Great Britain, which **[they]** do not know how to deal with, has strengthened in Ulster protestants.*

Open questions (3)

Nodes are reconstructed also with nominalizations:

*It [=the word] has a strong emotive **colouring** and it occurs especially in discourse of young people.*

colouring → Gen.ACT
→ Gen.PAT → *on*.PAT → *slovo* [word]

Work in progress (1)

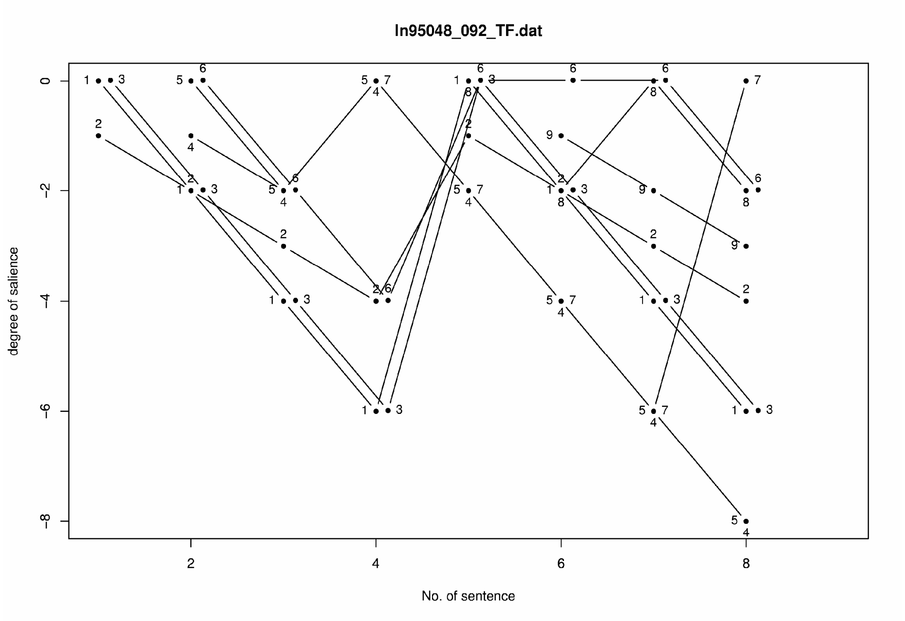
- Nouns as anaphors: anaphoric relations leading from nouns
- Rough classification
  - Identity
  - Part and whole relation
  - Function
  - Other types (of bridging anaphora)

Work in progress (2)

- Discourse structure analysis:
  - **Hypothesis:** A finite mechanism exists that enables the addressee to identify the referents on the basis of a partial ordering of the elements in the stock of knowledge (information) shared by the speaker and the addressees (according to the speaker's assumption), based on the degrees of activation of referents.

Stock of shared knowledge

- SSH: a structured whole
- Hierarchy of activation of the SSK elements – a partial ordering
- Heuristic rules" for the assignment of degrees of activation based on:
  - TFA value
  - coreferential links
  - outer form (pronoun, full noun group)
- Implementation of the rules and visualization of the results



Conclusions

- a systematic annotation of a large corpus of (segments of) continuous text(s) on several layers has an indisputable advantage
- there are, of course, many other respects in which corpus annotation schemes should go beyond the current practice
- there are no “frontiers” of the usefulness of annotated corpora both for linguistic theory and NLP applications

Part 4

DATA (continued): The Prague Dependency Treebank

The Prague Dependency Treebank and Valency Annotation (part 4)

Jan Hajič
Institute of Formal and Applied Linguistics
School of Computer Science
Faculty of Mathematics and Physics
Charles University, Prague
Czech Republic

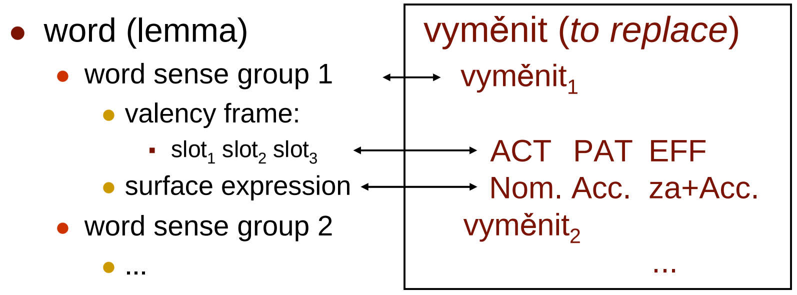
Prague Dependency Treebank Deep syntax & valency (part 4)

- Valency in the PDT
  - Valency lexicon for PDT
  - General valency lexicon
- Valency in deep vs. surface syntax
  - Links between the layers w.r.t. valency
- Valency and word sense
  - Sense-disambiguated occurrences:
    - Links from data to the lexicon
- Valency in translation, text generation

Definition of Valency

- Ability (“desire”) of words (verbs, nouns, adjectives) to combine themselves with other units of meaning
- Properties of valency:
  - Specific for every word meaning (in general)
    - leave: *sb left sth for sb* vs. *sb left from somewhere*
    - same as in PropBank *leave.02* vs. *leave.01*
  - Typically strongly correlates with surface form
    - morphological case (~ ending), preposition+case, ...
  - ~~Semantic constraints~~ ⚠ are very dangerous

Structure of Valency



- word (lemma)
  - word sense group 1
    - valency frame:
      - slot_1 slot_2 slot_3
    - surface expression
  - word sense group 2
    - ...

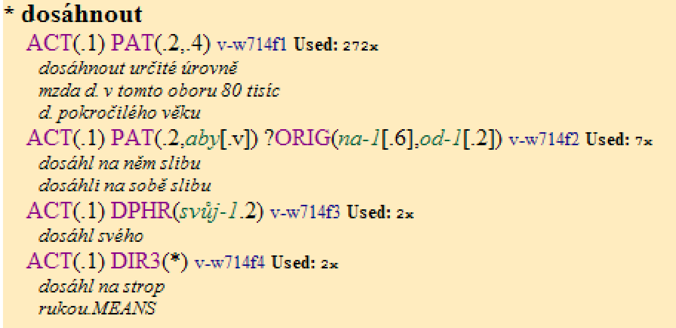
vyměnit (*to replace*)
vyměnit$_1$
ACT PAT EFF
Nom. Acc. za+Acc.
vyměnit$_2$
...

The Valency Lexicon PDT-VALLEX

- Valency frames
    - each verb, some nouns, adjectives
- Basic set prepared in advance, annotators add entries on-the-go, checking and approval process follows (consistency)
- VALLEX
    - more detailed and complex annotation of valency
    - Žabokrtský, Lopatková (2005), VALLEX 1.0
    - All about valency: http://ufal.ms.mff.cuni.cz/~semecky/vallex/

PDT-VALLEX Entry

- dosáhnout: *"to reach"*, *"to get [sb to do sth]"*
- browser/user-formatted example:



*** dosáhnout**

ACT(.1) PAT(.2,.4) v-w714f1 **Used:** 272x

dosáhnout určité úrovně

mzda d. v tomto oboru 80 tisíc

d. pokročilého věku

ACT(.1) PAT(.2,*aby*[.v]) ?ORIG(*na-1*[.6],*od-1*[.2]) v-w714f2 **Used:** 7x

dosáhl na něm slibu

dosáhli na sobě slibu

ACT(.1) DPHR(*svůj-1*.2) v-w714f3 **Used:** 2x

dosáhl svého

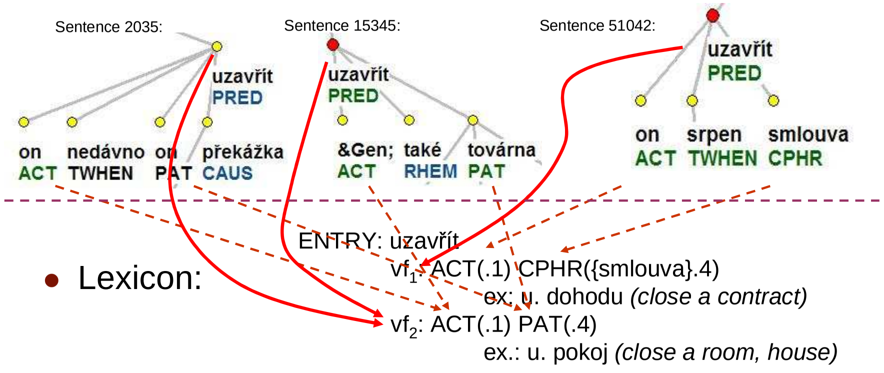
ACT(.1) DIR3(*) v-w714f4 **Used:** 2x

dosáhl na strop

rukou.MEANS

Corpus <-> Valency Lexicon

- Corpus:



- Lexicon:

ENTRY: uzavřít

vf_1: ACT(.1) CPHR({smlouva}.4)

ex.: u. dohodu *(close a contract)*

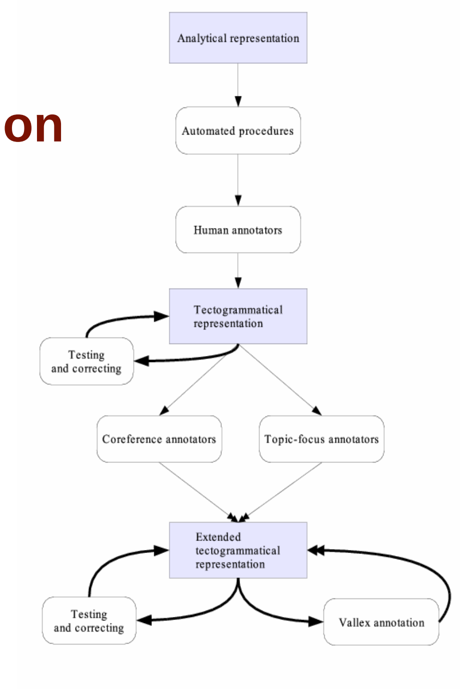
vf_2: ACT(.1) PAT(.4)

ex.: u. pokoj *(close a room, house)*

The Annotation Process

- 4 sublayers
    - work on structure first, rest in parallel
- Structure
    - automatic preprocessing - programmed conversion from analytical layer annotation
- Grammatemes
    - mostly automatically (based on lower layers' annotation), manual checking, corrections
- Cross-sublayer/cross-layer checking
    - partly automatic, then manual

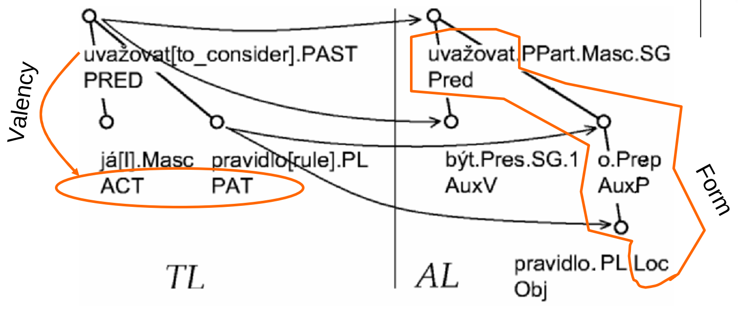
The Annotation Process Scheme



Valency & Tectogrammatical Annotation

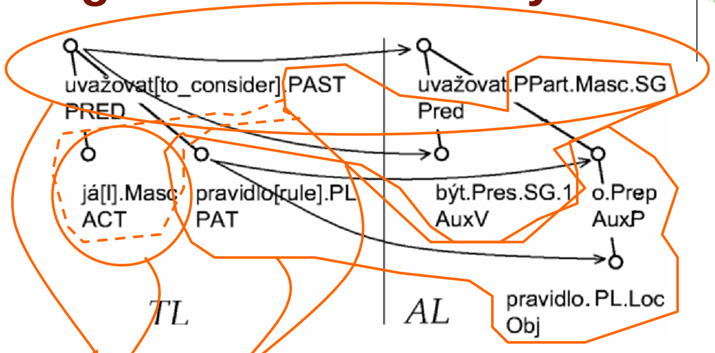
- Valency and...
  - (surface) form
- Annotation tools
  - TrEd
    - structural annotation
    - valency lexicon integration
- Search
  - TrEd, Netgraph

Valency & Form



lemma (AL): uvažovat
ACT: surface ellipsis, node disappears
PAT: preposition ‘o’ and a locative case

Tectogrammatical / Analytical



uvažovat – uvažovat
PAST / já.Masc – PPart.Masc.SG(Pred) / být.Pres.SG.1(AuxV)
pravidlo.PL.PAT – o.Prep(AuxP) / pravidlo.PL.Loc(Obj)
já - 0

?

<u>CONTEXT NEEDED</u>

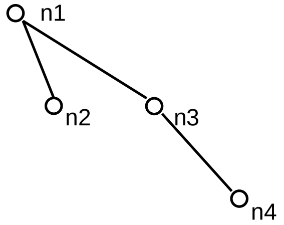
from another sentence: pravidlo.PL.PAT – pravidlo.PL.Acc(Obj)

Valency & Form

- Valency frame:
  - (per each sense of word)
  - (obligatory) modifiers ↔ functors
  - functor → form
- Simplest case:
  - surface form of a functor: particular case
  - Ex.: ACT in nominative (<u>he</u> says)
  - Ex.: PAT in accusative (she sees <u>him</u>)
- ... but it is not always so simple (as we have already seen)!

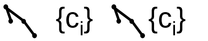
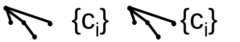
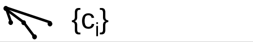
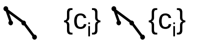
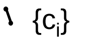
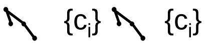
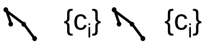
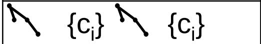
Valency & Form: Constraints

- Tree structure:



- (Sets of) Constraints:
  - n1: lemma=uvažovat mode=active
  - n2: case=Nom afun=Sb
  - n3: lemma=o afun=AuxP
  - n4: case=Loc afun=Obj

(General)
Valency Lexicon Entries

Entry	Sense #	Frame #	Valency Optinality		Form alternatives
1	1	1	ACT PAT		 $\{c_j\}$ $\{c_j\}$
	2	2	ACT PAT LOC		 $\{c_j\}$ $\{c_j\}$
		3	ACT PAT DIR3		 $\{c_j\}$
	3	4	ACT PAT		 $\{c_j\}$ $\{c_j\}$
2	1	1	ACT		 $\{c_j\}$
	2	2	ACT INTT		 $\{c_j\}$ $\{c_j\}$
3	1	1	ACT PAT		 $\{c_j\}$ $\{c_j\}$
	2	2	ACT PAT		 $\{c_j\}$ $\{c_j\}$

Valency Lexicon Simplification

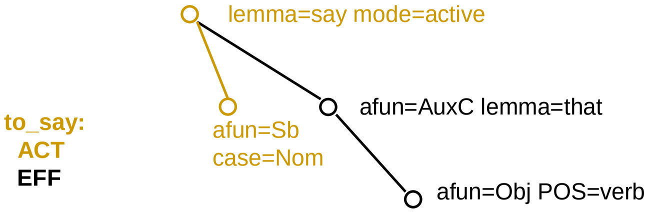
- Independent form for each slot of a particular valency frame
  - ACT, PAT, ...: own constraint, not a global one
- $\text{Functor}_{\text{oblig./opt.}}$ ↔ $\text{constraints}_{\text{Functor}}$
- Ex.:
  - lemma1 ACT(Nom.) PAT(o+6) (to consider a rule)
  - lemma2 ACT(Nom.) PAT(4) (create a rule)
- Standard “transformations” of frame form
  - passivization, reflexivization, ...

Example: Valency & Form

- Simple 1:1:
  - ex.: create: ACT(Nom) PAT(Acc)
  - verb in infinitive: INTT(Inf)
  - subordinate clause: PAT(verb)
  - class of words with generic verbs: CPHR({class})
  - no constraint: (often) LOC, TWHEN
    - general constraint for a given functor applies
  - ...more!

Example: Valency & Form

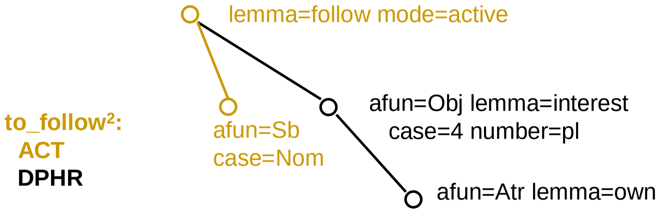
- 1:2
  - relative clause



- linear representation: EFF(that[.v])

Example: Valency & Form

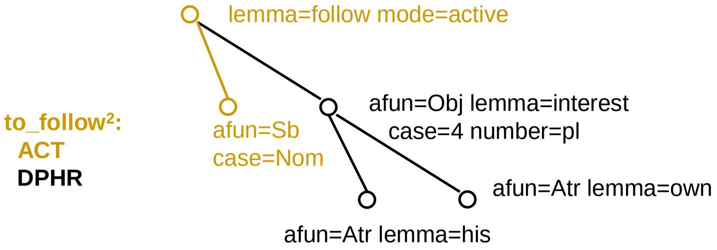
- 1:2
  - idomatic phrase



- linear representation: DPHR(interest.P4[own.#])

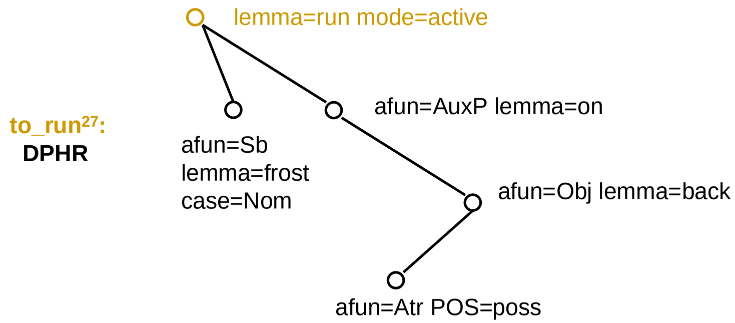
Example: Valency & Form

- 1:3
  - idomatic phrase



Example: Valency & Form

- 1:4
  - idomatic phrase



to_run^{27}:
DPHR

lemma=run mode=active

afun=Sb
lemma=frost
case=Nom

afun=AuxP lemma=on

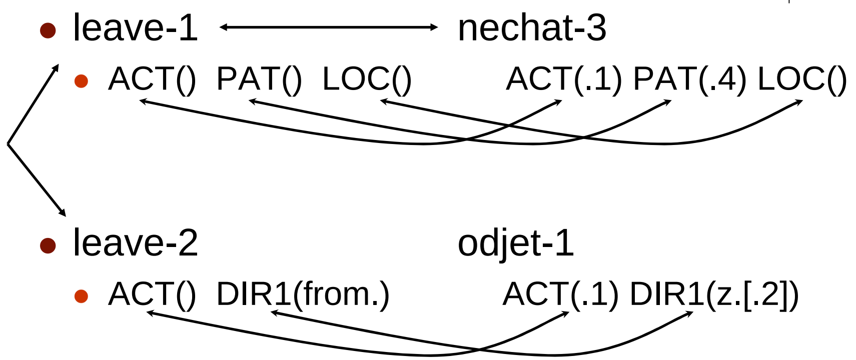
afun=Obj lemma=back

afun=Atr POS=poss

Valency and Translation

- leave:
  - leave-1
    - to leave [from] somewhere
  - leave-2
    - to leave sth for sb
- Translating (from English into Czech):
  - which equivalent to chose?
    - nechat vs. odjet/opustit
  - which prepositions, cases, ... to use?
    - accusative vs. “z” (*“from”*) with genitive vs. ...?

Valency and Translation



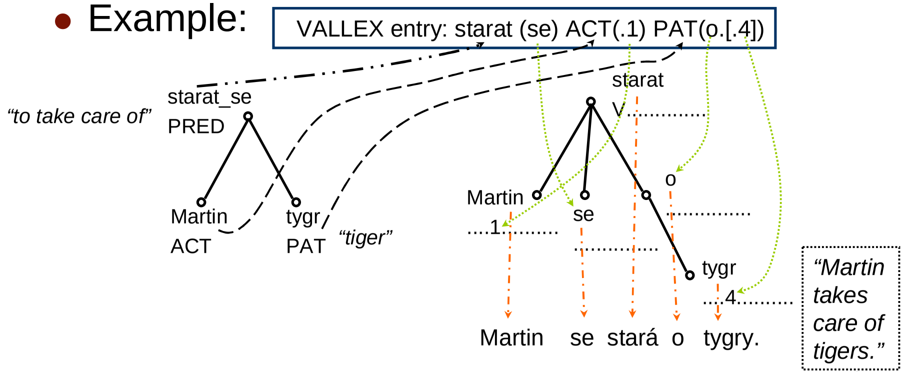
- leave-1 ⟷ nechat-3
  - ACT() PAT() LOC() ACT(.1) PAT(.4) LOC()
- leave-2 odjet-1
  - ACT() DIR1(from.) ACT(.1) DIR1(z.[.2])

Valency and Text Generation

- Tectogrammatical Representation
  - has *__all__* the information to (re)generate the surface form of the sentence:
    - in a “generalized” form
    - non-redundant (almost... but for generation, it is o.k.)
  - ...except the links to a-layer, however
    - links used only for *__training__* [statistical models for] parsing/generation modules
    - not present when e.g. doing text planning, translation, ...
  - valency dictionary: form of “learned” knowledge

Valency and Text Generation

- Using valency for...
  - ...getting the correct (lemma, tag) of verb arguments
- Example:



Tectogrammatical Annotation Tools

- Manual annotation
  - 4 groups of annotators ~ 4 sublayers
  - Special graphical tool (TrEd)
    - Customizable graphical tree editor
- Preprocessing
  - Data from analytical layer, preprocessed
  - Online dependency function preassignment

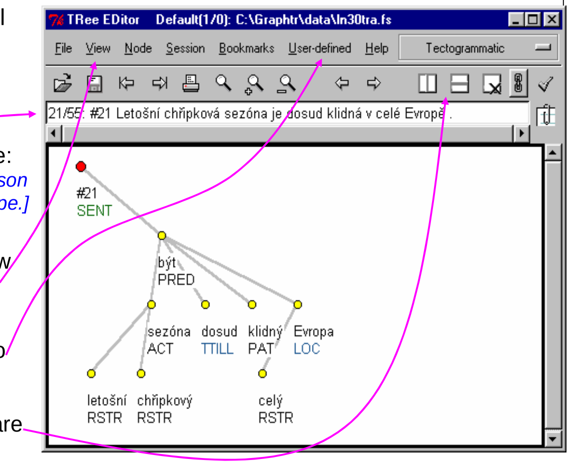
The [Manual] Annotation Tool

- Perl/PerlTk based, platform-independent
  - Linux, Windows 95/98/2000, Solaris, ...
- Perl as the “macro” language
  - “unlimited” online processing capability
- Flexibility for interactive checking
  - split screen, graphical “diff” function
- Customization, printing, “plugins”, ...
- !! See also J. Stepanek’s lecture / tools

The “TrEd” Tree Editor

- Graphical tool

  TrEd
- Main screen:



21/55: #21 Letošní chřipková sezóna je dosud klidná v celé Evropě .

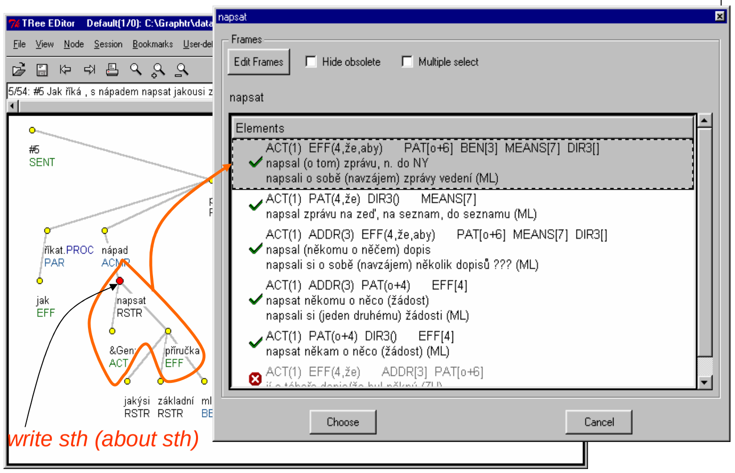
Original sentence:
[This year’s flu season is still quiet in Europe.]

Editing window customization

Run a macro

Multiwindow editing/compare

Valency Lexicon in TrEd



5/54: #5 Jak říká , s nápadem napsat jakousi z

☐ Hide obsolete ☐ Multiple select

napsat

✓ ACT(1) EFF(4,že,aby) PAT[o+6] BEN[3] MEANS[7] DIR3[]
napsal (o tom) zprávu, n. do NY
napsali o sobě (navzájem) zprávy vedení (ML)

✓ ACT(1) PAT(4,že) DIR3() MEANS[7]
napsal zprávu na zeď, na seznam, do seznamu (ML)

✓ ACT(1) ADDR(3) EFF(4,že,aby) PAT[o+6] MEANS[7] DIR3[]
napsal (někomu o něčem) dopis
napsali si o sobě (navzájem) několik dopisů ??? (ML)

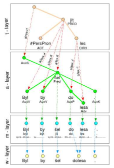
✓ ACT(1) ADDR(3) PAT(o+4) EFF[4]
napsat někomu o něco (žádost)
napsali si (jeden druhému) žádosti (ML)

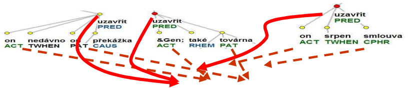
✓ ACT(1) PAT(o+4) DIR3() EFF[4]
napsat někam o něco (žádost) (ML)

☒ ACT(1) EFF(4,že) ADDR[3] PAT[o+6]

to write sth (about sth)

Annotating the Links





- Stand-off annotation principles
  - Links to another layer
  - Links to lexicon
- Minimal work on link annotation (close to zero)
- Macro commands in TrEd
  - transparently keeps track of merged nodes, splits, etc., and adapts links correspondingly.
- Result:
  - almost no extra work
  - final check after annotators do the last pass

The “Old” PDT 1.0

- Morphology (1.8MW) & Surface syntax (1.5MW)
- SGML format (csts.dtd) + compact “FS”
- Mixed (single-file) annotation
  - 7 attributes + dependency
- TrEd (graphical viewer/editor), NetGraph (search capability)
  - simple visualization

What’s New in PDT 2.0

- Tectogrammatical layer (0.8MW)
  - 39 node attributes + dependency
  - valency dictionary (PDT-VALLEX)
- XML stand-off annotation (“PML”, 4 layers)
- New data division (train/dtest/etest)
  - added morphological annotation to all data
  - corrections of PDT 1.0 files (morphology, syntax)
- Improved tools:
  - TrEd, btred/ntred (batch tree corpus processing)
    - new features, better visualization

Tectogrammatical attributes I

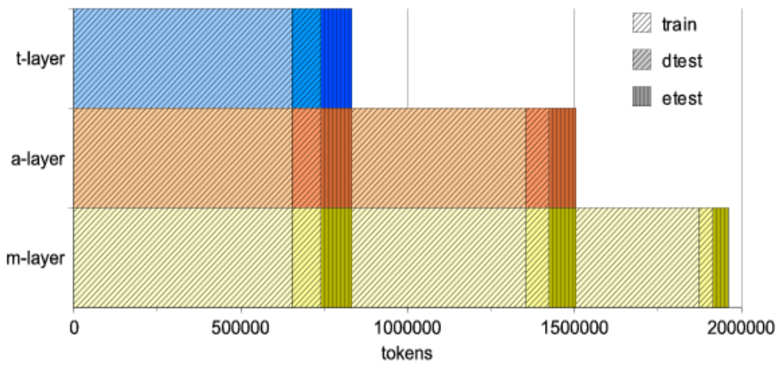
- node typing
  - complex, coap, qcomplex, root, atom, ...
- functor, subfunctor
  - TWHEN: TWHEN.basic, TWHEN.before
- is_member, is_generated, is_parenthesis, is_dsp_root, is_state, quot_type, ...
- grammatemes (16):
  - aspect, degcmp, deontmod, sempos, tense, indeftype, politeness, person, ...

Tectogrammatical attributes II

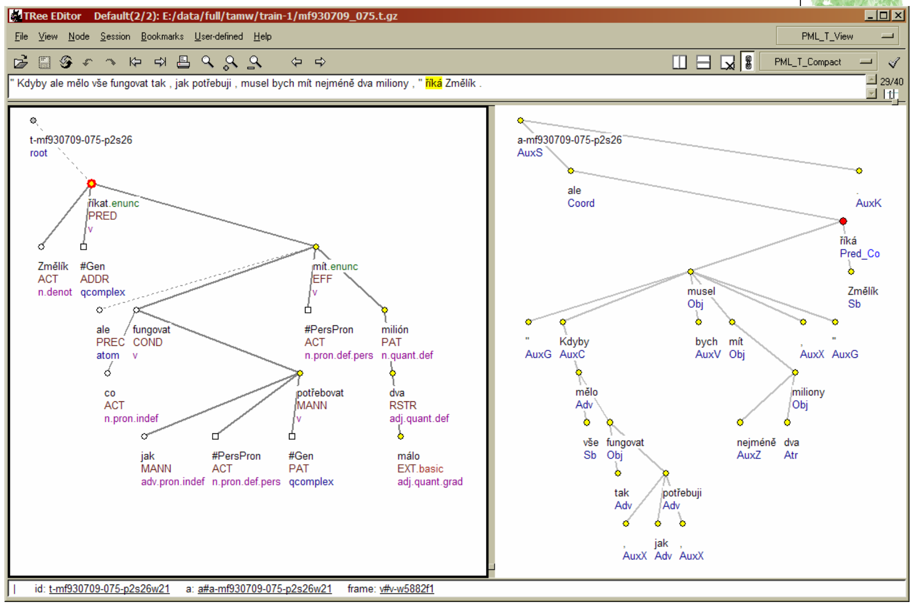
- topic/focus:
  - tfa, deepord
- valency: t_lemma, val_frame.rf
- bookkeeping: id
- coref_gram.rf, coref_text.rf, compl.rf
  - reference to TR node, type of coreference
- sentmod
- Linking to analytical layer
  - a.lex.rf (“main” anal. node), a.aux.rf (others)

PDT 2.0: The Data

- Data sizes



TrEd



" Kdyby ale mělo vše fungovat tak , jak potřebuji , musel bych mít nejméně dva miliony , " říká Změlík .

id: t-mf930709-075-p2s26w21 a: a#a-mf930709-075-p2s26w21 frame: v#v-w5882f1

Using the Results (t-layer)

- Preliminary!
  - PDT 2.0 published July 2006
  - 50k sentences for training (t-layer)
- Functor assignment
  - > 80% accuracy on manually annotated structure
- Tectogrammatical parser
  - Part of the “toolchain” (run_all, see p. 5, p. 7, J. Štěpánek)
- Coreference
  - preliminary results: > 80%
- Valency
  - frame assignment > 70%

To take home...

- What is PDT
  - Dependency-based treebank project
    - Czech (other languages in the works)
  - ~ 1mil. words
    - sufficient size for ML experiments
  - 4 layers of annotation
    - token, morphology, syntax, deep syntax/semantics++)
    - independent and full information at all levels, but...
    - interlinked (for the development of parsers/generators)
  - Valency dictionary integrated (links from data)

Some (more) pointers

- http://ufal.mff.cuni.cz/pdt2.0
  - Current version of PDT, all three levels, 1.9/1.5/0.8 Mw
- http://ufal.mff.cuni.cz/REST/CAC/CAC.html
  - The Czech Academic Corpus, v 1.0
- http://www.ldc.upenn.edu
  - LDC2001T10 (PDT v1.0), LDC2004T23 (PADT 1.0), LDC2004T25 (PCEDT 1.0)
- http://www.clsp.jhu.edu: Workshop 2002
  - Using TL for MT Generation

Part 5

TOOLS:
Annotation editors
Browsers and viewers

Lexical Annotation Workbanch
LAW

Integrated environment
for morphological desambiguation
operating on m-layers

Processing

- Input - morphologically annotated and lemmatized text (from automatic analysis)
  - several (many) possibilities for each token
- Manual selection of the right possibility
  - either from given options
  - or addition of a new option
- Output - morphologically annotated and lemmatized text - desambiguated
    - allowed more possibilities

Filters

- only ambiguous
- only selected tags (regular expressions)
- only selected lemmas (regular expressions)
- only selected words (regular expressions)
- continuous filtering possible
- comparing two different m-layers

Prague Treebanking for Everyone

Annotation Editor for the Analytical and Tectogrammatical Layer and Valency Lexicon

Jan Štěpánek

29th November 2006

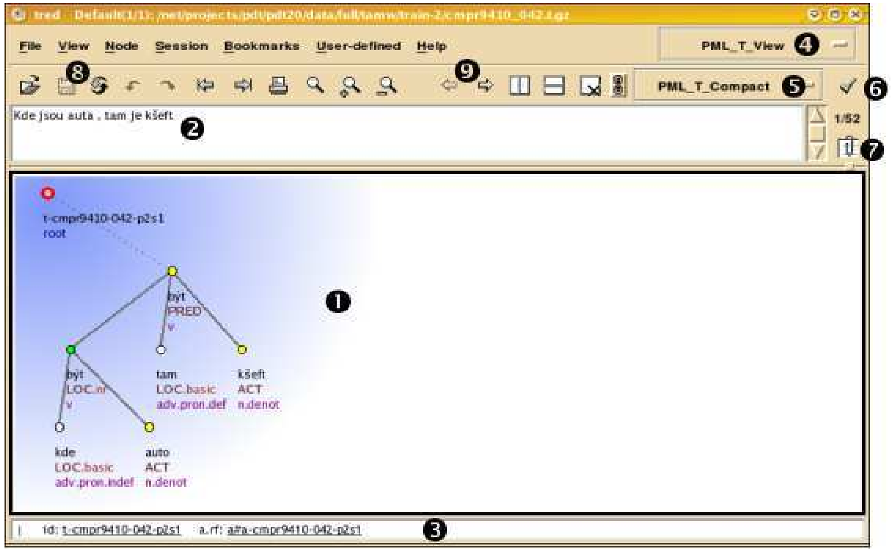
Tree Editor TrEd

- Independent on operating system (MS Windows, Linux, OS X...)
- Open source program, available for free
- Written in **perl** (macros, predefined functions)

Requirements and Installation

- **perl** (5.8.3 or newer) with **Tk** library
- `http://ufal.mff.cuni.cz/~pajas/tred/`
  - For MS Windows: `tred_wininst_en.zip` → `setup.bat`
  - For Linux: `tred-dep-unix.tar.gz` → install `tred-current.tar.gz`

TrEd Window



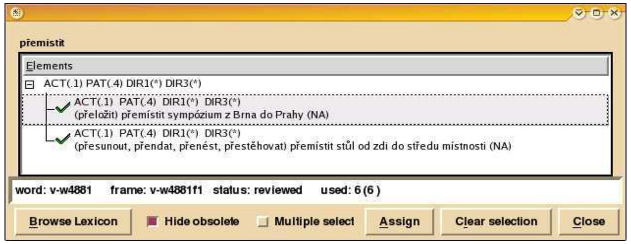
Kde jsou auta , tam je kšeft

id: t-cmpr9410-042-p2s1 a.rf: a#a-cmpr9410-042-p2s1

TrEd Window (2)

1. Main frame(s). Each frame displays one tree.
2. A textual form of the sentence of the current tree.
3. Status line – various information.
4. Current context (set of macros). Can be changed by clicking and selecting a new one.
5. Current stylesheet.
6. “Edit stylesheet” button.
7. Position of the current sentence in the file; “show sentences” button.
8. Buttons to open, save and reload a file. The icons mean Undo, Redo, Previous and Next File, Print, Find, Find Next, Find Previous.
9. Buttons for moving to the previous/next tree in the current file and for frame management.

Browsing Valency Lexicon



přemistit

ACT(.1) PAT(.4) DIR1(*) DIR3(*)

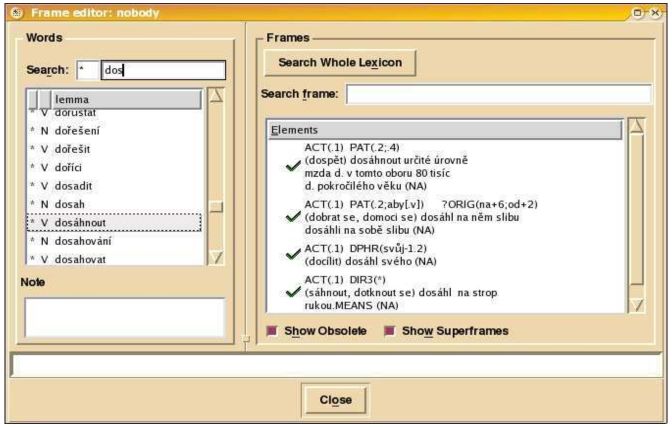
ACT(.1) PAT(.4) DIR1(*) DIR3(*)
(přeložit) přemístit sympózium z Brna do Prahy (NA)

ACT(.1) PAT(.4) DIR1(*) DIR3(*)
(přesunout, přendat, přenést, přestěhovat) přemístit stůl od zdi do středu místnosti (NA)

word: v-w4881 frame: v-w4881f1 status: reviewed used: 6 (6)

`Ctrl+Enter`: ValLex entry for the current node

Browsing Valency Lexicon



Search: * dos

Search frame:

* V dorustat
* N dořešení
* V dořešit
* V dořici
* V dosadit
* N dosah
* V dosáhnout
* N dosahování
* V dosahovat

ACT(.1) PAT(.2;.4)
(dospět) dosáhnout určité úrovně
mzda d. v tomto oboru 80 tisíc
d. pokročilého věku (NA)

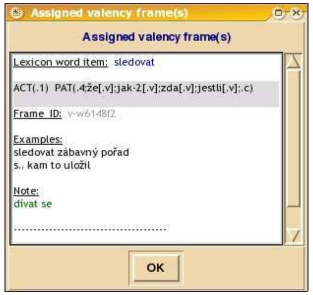
ACT(.1) PAT(.2;aby[.v]) ?ORIG(na+6;od+2)
(dobrat se, domoci se) dosáhl na něm slibu
dosáhli na sobě slibu (NA)

ACT(.1) DPHR(svůj-1.2)
(docílit) dosáhl svého (NA)

ACT(.1) DIR3(*)
(sáhnout, dotknout se) dosáhl na strop
rukou.MEANS (NA)

`Ctrl+Shift+Enter`: Browsing the ValLex

Browsing Valency Lexicon



Lexicon word item: sledovat

ACT(.1) PAT(.4;že[.v];jak-2[.v];zda[.v];jestli[.v];.c)

Frame ID: v-w6148f2

Examples:
sledovat zábavný pořad
s., kam to uložil

Note:
dívat se

`Ctrl+v`: Show the assigned valency frame

Corpus manager Manatee / Bonito

- Searching in corpora according to all attributes or their combinations (regular expressions)
- Sorting and filtering results
- Basic statistics
- Graphical overview of occurences within the whole corpus
- Display of more attributes (word, lemma, different types of tags, ...)
- Creating subcorpora

Searching

- most common attributes:
  - word
  - lemma
  - morphological tag
  - (analytic functors)
- Regular expressions
- It is possible to combine the attributes

[tag="R.+"][lemma="Praha.*"]

Displaying

- Selection of attributes
  - for KWIC
  - for all positions
- Changing range - number of concordances
  - from beginning / end / middle
  - randomly
- Information about the source of the text
- Changing context
- Wider context in extra region

Sorting and Filtering

- Sorting according to
  - kwic (key word in context)
  - left or right context
  - combination
- Filters
  - positive
  - negative

Statistics

- Frequency distribution
  - according to selected attributes
- Collocations
  - absolute frequency
  - relative frequency
  - Mutual information
  - t- score
  - distribution overview + Average Reduced Frequency

$$mi(x,y) = \log_2 \frac{N \cdot f(x,y)}{f(x) \cdot f(y)}$$

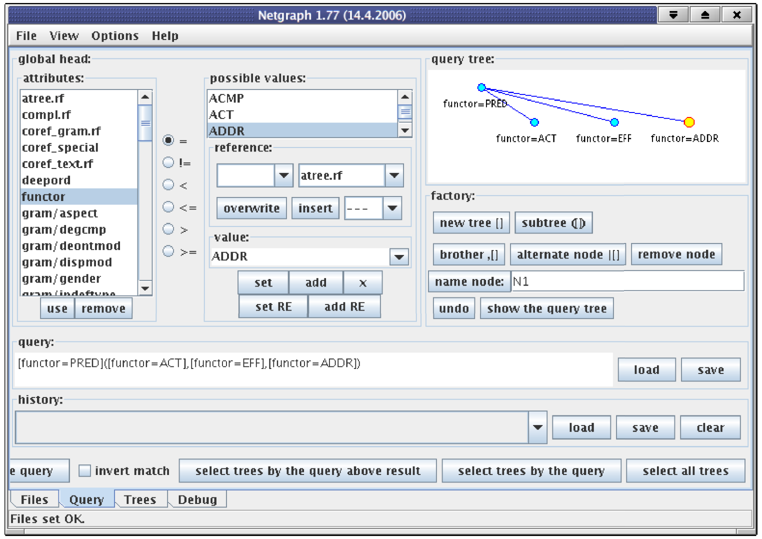
$$T = \frac{\left(f(x,y) - \frac{f(x)\,.\,f(y)}{N}\right)}{\sqrt{f(x,y)}}$$

$$ARF = 1/v \sum_{i=1}^{f} \min\{d_i, v\}$$

Netgraph – a Tool for Searching in Prague Dependency Treebank 2.0

- Client-server architecture
- Authentication of users
- Subcorpus definition
- Graphic creation of a query
- Searching in the treebank according to the query
- Viewing the result trees
- Basic statistics

A Query Creation



value: ADDR

name node: N1

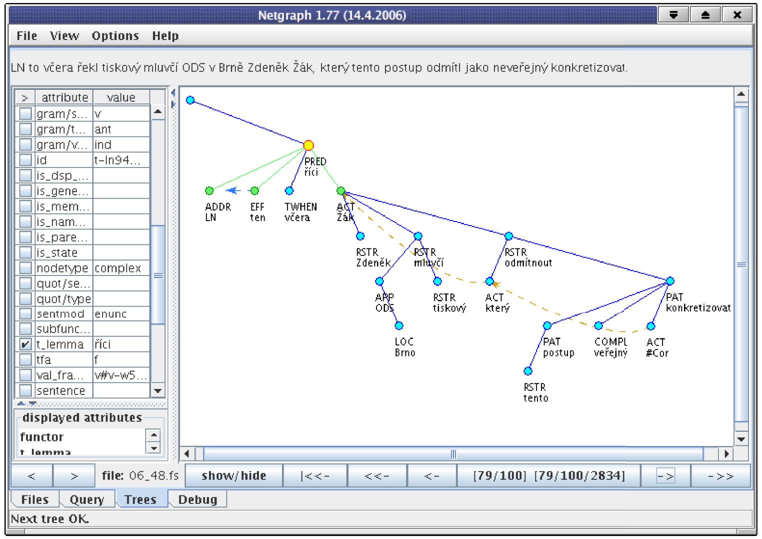
query: [functor=PRED]([functor=ACT],[functor=EFF],[functor=ADDR])

☐ invert match

Files set OK.

[functor=PRED]([functor=ACT],[functor=EFF],[functor=ADDR])

Viewing the Result



LN to včera řekl tiskový mluvčí ODS v Brně Zdeněk Žák, který tento postup odmítl jako neveřejný konkretizoval.

>	attribute	value
☐	gram/s...	v
☐	gram/t...	ant
☐	gram/v...	ind
☐	id	t-ln94...
☐	is_dsp_...	
☐	is_gene...	
☐	is_mem...	
☐	is_nam...	
☐	is_pare...	
☐	is_state	
☐	nodetype	complex
☐	quot/se...	
☐	quot/type	
☐	sentmod	enunc
☐	subfunc...	
☑	t_lemma	říci
☐	tfa	f
☐	val_fra...	v#v-w5...
☐	sentence	

file: 06_48.fs [79/100] [79/100/2834]

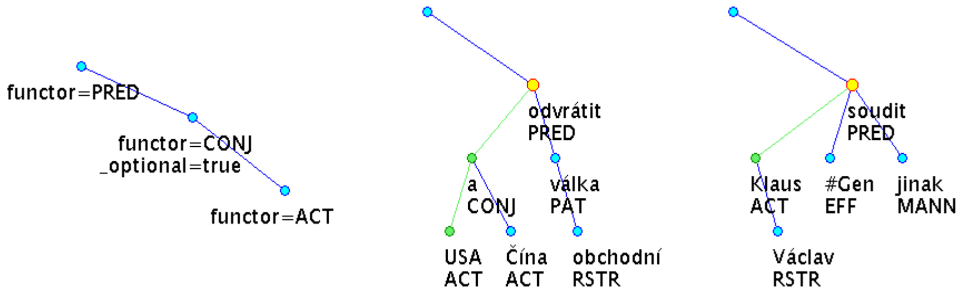
Next tree OK.

- Different order of nodes; additional sons of the PREDicate

Meta-attributes

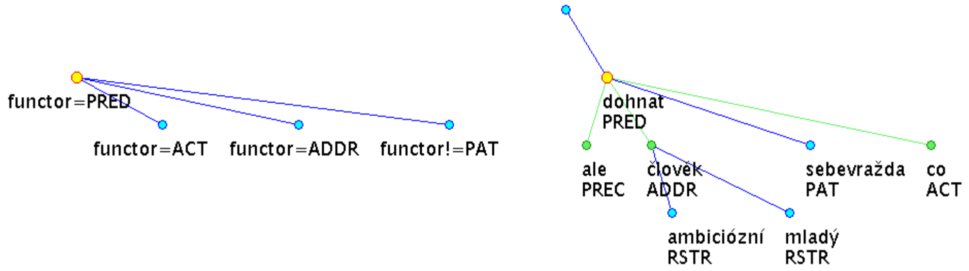
- Additional power to the query language
- Attributes not present in the corpus
- Treated like normal attributes
  - **_transitive** *(transitive edge)*
  - **_optional** *(optional node)*
  - **_#sons** *(exact number of sons)*
  - **_depth** *(distance from the root)*
  - ...

An Example Query



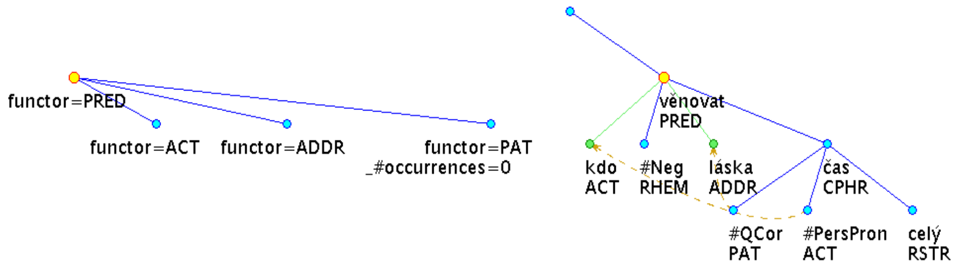
- A query with optional CONJunction node
- Two possible types of result – with and without the optional node

An Example of a Wrong Query



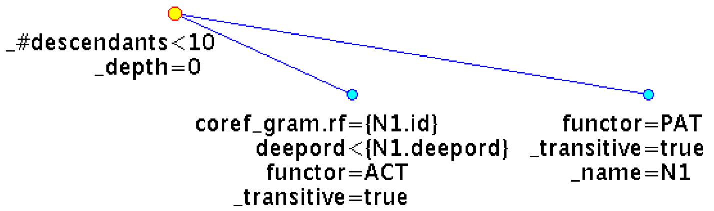
- A wrong attempt to set negation in the query
- We do not want the PATient there at all
- But the query node matches with PREC

A Correct Negation



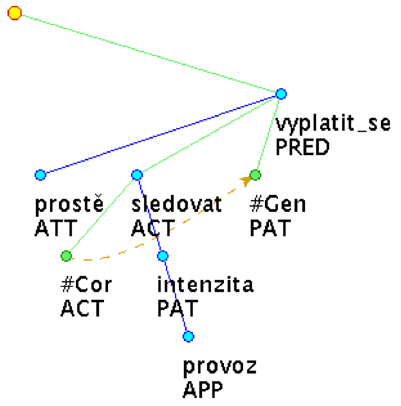
- A correct way how to set negation in the query
- We define that there are exactly zero PATients as sons of the PREDicate

Yet Another Example Query



- Looking for a small tree (root of the query)
- PATient is a coreferencial node of ACTor and is on the left side from the ACTor

A Result Tree



http://quest.ms.mff.cuni.cz/netgraph

Part 6

DATA:
The Prague Mark-up Language

XML-Based Format of PDT 2.0 Data

Petr Pajas

ÚFAL, MFF, Charles University, Prague

November 29, 2006

Introduction

In PDT 2.0 and some related annotation projects, we:

- use several (interlinked) annotation levels
- introduce various types of annotation: linear, tree-structured, . . .
- apply one annotation schema to different languages
- sometimes use project-specific variants/flavors of existing annotations or annotation schemas
- use something called *“annotation dictionaries”*

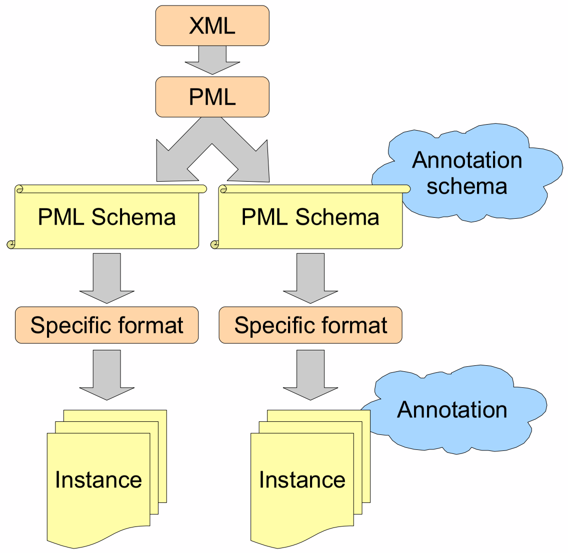
Our goal is to represent all data resulting from these projects coherently, in a uniform manner.

- We need a meta-format (from which we may derive specific formats for our annotation schemas and layers)
- XML is known to be an excellent meta-format, but still too generic to ensure uniformity of representation
- thus, we need another layer of abstraction, between the *“generic”* (XML) and the *“specific”* (data-specific format)

Requirements

- Uniformity of representation
- Stand-off annotation principles
- Unified cross-referencing and linking system
- Linearity and structure
- Structured attributes
- Handling ambiguity
- Human-readability
- Extensibility
- XML based
- Description language

PML – Prague Markup Language



PML – Prague Markup Language (2)

Every annotation layer has a PML-based format specified by something called *PML schema*.

- PML schema
  - *A data format description language.*
  - *Formalizes the "annotation schema" for a particular layer*
  - *Defines the annotation structure*
  - *Assigns PML roles to certain pieces of annotation.*
- annotation structure
  data structure built from abstract data types, such as: atomic values, attribute-value structures, lists, alternatives, etc.
- PML role
  identifies a piece of annotation as a bearer of some additional higher-level property, such as being a node of a tree, being a unique identifier, etc.

XML documents conforming to a PML schema are called PML instances.

PML - Prague Markup Language (3)

Processing PML

PML instances can be processed using:

- arbitrary XML-oriented tools (based on DOM, XPath, etc.)
- format-specific tools with hard-wired knowledge about the XML vocabulary of a particular PML-based format
- intelligent generic tools aware of PML-schema:
  - data type declarations → optimal in-memory representation (data binding)
  - role assignment → adequate way of presenting the annotation to the user and providing some extra features (indexes, etc.)

Validation

- using conventional validators for XML such as `xmllint` or `jing`. (PML schema translates to a Relax NG – via XSLT)

Data types – overview

- **Atomic (simple)**
  - cdata
  - enumerated
  - constant
- **Complex**
  - structures
  - containers
  - lists
  - alternatives
  - sequences

Data types – atomic (1)

cdata type

- literal character-based content
- no explicitly marked internal structure
- value format can be restricted
- current formats include most W3C Schema simple types

PML schema declaration

```
<cdata format="token"/>
<cdata format="float"/>
<cdata format="positiveInteger"/>
<cdata format="long"/>
<cdata format="date"/>
<cdata format="time"/>
<cdata format="any"/>
...
```

→

instance example

```
<...>hallo234</...>
<...>12.7843E-2</...>
<...>17</...>
<...>-9223372054775808</...>
<...>1999-05-31</...>
<...>13:20:00.000</...>
<...>ar6!7rar¥ d@t&</...>
...
```

Data types – atomic (2)

enumerated type

Literal values from a fixed finite set.

PML schema declaration

```
<type name="boolean.type">
  <choice>
    <value>TRUE</value>
    <value>FALSE</value>
  </choice>
</type>
```

→ instance example

```
<...>TRUE</...>
```

constant

A fixed constant value.

PML schema declaration

```
<type name="root-node.type">
  <structure>
    <member name="node-type">
      <constant>root</constant>
    </member>
    ...
  </structure>
</type>
```

→ instance example

```
<...>
  <!-- no need to use explicitly -->
  ...
</...>
```

Data types – complex (1)

Attribute-value structure (AVS)

- consists of a fixed set of attribute-value pairs (called members)
- values can be atomic or complex
- value type of each member is declared in the PML schema
- members can be optional or required
- atomic-valued members can be rendered as attributes

PML schema declaration

```
<structure>
  <member name="id" as_attribute="1">
    <cdata format="ID" role="#ID"/>
  </member>
  <member name="form" required="1">
    <cdata format="token"/>
  </member>
  <member name="lemma">
    <cdata format="token"/>
  </member>
  <member name="tag"
       type="tagset.type"/>
</structure>
```

→ example instances

```
<... id="m-23">
  <form>walking</form>
  <lemma>walk-1</lemma>
  <tag>VBG</tag>
</...>

<...id="m-24">
  <form>away</form>
</...>
```

Data types – complex (2)

container

- attaches attributes to a single central value (content)
- attributes are name-value pairs with atomic values
- the set of attributes is declared in the schema
- content can be atomic or complex other than container or structure

PML schema declaration

```
<type name="word.type">
  <container>
    <attribute name="id"
            role="#ID">
      <cdata format="ID"/>
    </attribute>
    <cdata format="token"/>
  </container>
</type>
```

→ instance example

```
<... id="w-23">Walking</...>
```

Data types – complex (3)

list

- aggregate zero or more values of a certain type
- can be ordered or unordered (sets).
- reserved tag <LM> used for list values
- single <LM> can sometimes be ommitted (list folding)

PML schema declaration

```
<type name="sent.type">
  <list ordered="1" type="word.type">
</type>
```

→ example instance (several values)

```
<...>
  <LM id="w-34">Flies</LM>
  <LM id="w-35">like</LM>
  <LM id="w-36">an</LM>
  <LM id="w-37">arrow</LM>
  <LM id="w-38">.</LM>
</...>
```

→ example instance (single value)

```
<...>
  <LM id="w-34">Flies</LM>
</...>
```

can fold into:

```
<... id="w-34">Flies</...>
```

Data types – complex (3)

alternative

- aggregates values in parallel, i.e. as alternative annotations.
- reserved tag `<AM>` used for alternative values
- single `<AM>` can be ommitted (folding)

PML schema declaration

```
<type name="morph.type">
  <alt type="m.type">
</type>
```

→

instance example

```
<...>
  <AM id='m-34'>
    <form>flies</form>
    <lemma>fly-1</lemma>
    <tag>VBZ</tag>
  </AM>
  <AM id='m-34A'>
    <form>flies</form>
    <lemma>fly-2</lemma>
    <tag>NNS</tag>
  </AM>
</...>
```

Data types – complex (4)

sequence

- aggregates values of several different types
  (unlike lists which are type-homogeneous)
- consists of zero or more name-value pairs (called *elements*)
- element's name determines its value type
- elements may be arbitrarily repeated
- reg.exp.-like pattern may restrict element order

PML schema declaration

```
<type name="chapter.type">
  <sequence
      content_pattern="para*, sect+">
    <element name="para"
        type="paragraph.type"/>
    <element name="sect"
        type="section.type">
  </sequence>
</type>
```

→

instance example

```
<...>
  <para>
    In this chapter...
  </para>
  <sect>...</sect>
  <sect>...</sect>
  <sect>...</sect>
</...>
```

PML roles

- ID

  assigned to members or attributes uniquely identifying an AVS or a container within a PML instance
- KNIT

  assigned to links suitable for merging two annotation layers in such a way that the referred object is embedded into the referring object
- TREES

  marks lists or sequences of dependency or constituency trees
- NODE

  identifies nodes of dependency or constituency trees
- CHILDNODES

  identifies the member of a node (of a dependency or constituency tree) containing the list of its child nodes
- ORDER

  used to identify numerical values which determine a total ordering on a tree

---

Links

- Currently only ID-based links are supported
- Other types of links represented in PML on a per-application basis

ID-based links:

- Cross-references within a single PML instance
- Links to other instances

  Typically many links to only few target instances

  Therefore PML links have two parts:
  - the specification of the target instance – a label (ID) associated with the target instance in the referring instance header
  - the ID of the target object

Links - examples

Associating target URLs with IDs in the referring instance header

```
<references>
  <reffile id="a" href="doc73.a"/>
  <reffile id="v" href="http://mysite/vallex.xml"/>
</references>
```

Examples of ID-based links

```
<coref.rf>t-node-232</coref.rf>        — link to the same file

<val_frame.rf>v#f2234</val_frame.rf>  — link to http://mysite/vallex.xml

<lex.rf>a#doc73-w5</lex.rf>            — links to doc73.a

<aux.rf>
  <LM>a#doc73-w3</LM>
  <LM>a#doc73-w4</LM>
</aux.rf>
```

Links to non-PML data

Currently no guidelines.

Example of possible PML representation of pointers to an audio file:

```
<references>
  <reffile id="au1" href="spk1_129.ogg"/>
</references>
...
<w id="w-12941">
  <token>_SIL_</token>
  <audio>
    <time_start>600000</time_start>
    <time_end>4700000</time_end>
    <file.rf>au1</file.rf>
  </audio>
</w>
```

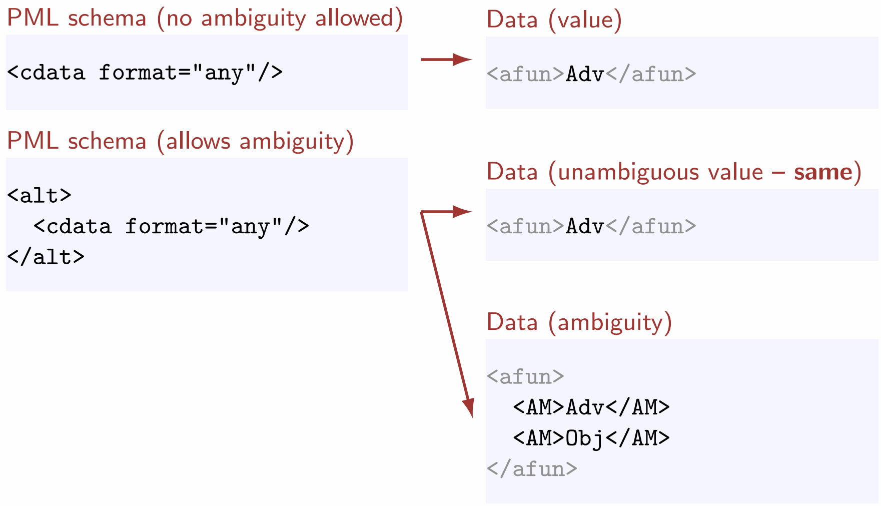
Extendibility

PML allows to upgrade or extend the data model in many ways while retaining the XML representation of the existing data.

- alternative folding
- list folding
- container transparency
- structure to sequence conversion
- structure and sequence extendibility

Extendibility – singleton alternative minimization

Data fields of any type may be upgraded to allow ambiguity:



PML schema (no ambiguity allowed)

```
<cdata format="any"/>
```

Data (value)

```
<afun>Adv</afun>
```

PML schema (allows ambiguity)

```
<alt>
  <cdata format="any"/>
</alt>
```

Data (unambiguous value – **same**)

```
<afun>Adv</afun>
```

Data (ambiguity)

```
<afun>
  <AM>Adv</AM>
  <AM>Obj</AM>
</afun>
```

Extendibility – singleton list minimization

By the „uniformity of representation“ principle, we get the same for lists.

Upgrading a value to a lists of values:

PML schema (one-to-one link)

```
<cdata format="PMLREF"/>
```

Data (link)

```
<m.rf>m-34</m.rf>
```

PML schema (one to many)

```
<list>
  <cdata format="PMLREF"/>
</list>
```

Data (folded singleton list)

```
<m.rf>m-34</m.rf>
```

Data (multiple values)

```
<m.rf>
  <LM>m-34</LM>
  <LM>m-34A</LM>
</m.rf>
```

Extendibility – container transparency

Add attribute annotation to flat data, lists, etc.

PML schema (flat value)

```
<cdata format="string"/>
```

Data (flat value)

```
<phrase>in the red</phrase>
```

PML schema (annotated value)

```
<container>
  <attribute name="category"
        type="phr-category.type"/>
  <attribute name="area"
        type="phr-area.type"/>
  <cdata format="string"/>
</container>
```

Data (container w/o attributes)

```
<phrase>in the red</phrase>
```

Data (adding attributes)

```
<phrase category="idiom" area="financial">in the red</phrase>
```

Modularization

Allows to easily derive one PML schema from another.

- revision numbering

  PML schemas can be assigned revision numbers of the form $X.Y.Z\ldots$

- `<import>` instruction

  Copies type declarations from a different PML schema with revision restrictions.

- `<derive>` instruction

  Derive types from a previously declared or imported AVS structure, container, sequence, or enumerated type.

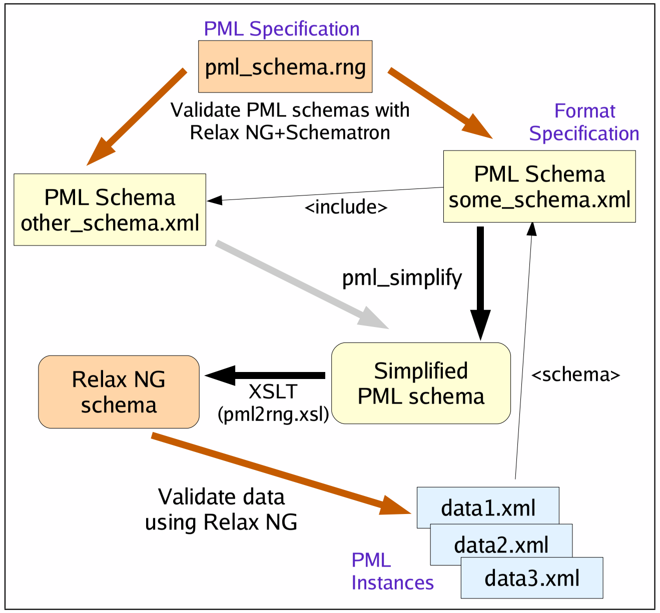
- simplified PML schemas

  `pml_simplify` *- a PML schema preprocessor, resolves all* `<import>` *and* `<derive>` *instructions*

  Useful e.g. before XSLT 1.0 transformations.

Validation

- Conformance of PML schemas to the spec can be verified via Relax NG and Schematron.
- Simplified PML schemas can be XSLT-transformed into Relax NG for data validation purposes.
- Many validators for Relax NG exist.
- The script `validate_pml` automates all the necessary steps.



Why Relax NG + Schematron

...and not W3C XML schemas or DTD?

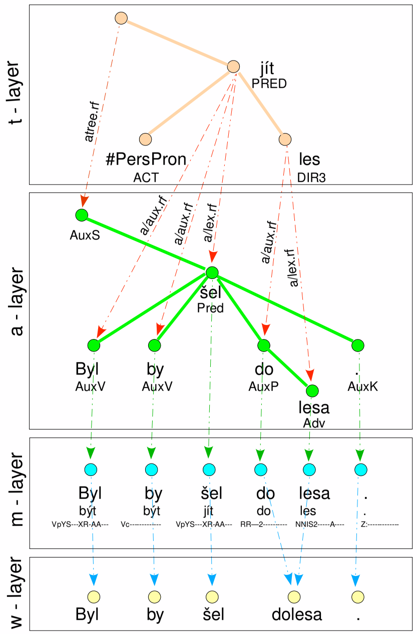
- First, DTD? Really? You must be kidding, right?
- W3C schemas support attributes as poorly as DTDs.
- Relax NG has way more flexible structural support than W3C schemas.
- Extendibility of W3C schemas is questionable.
- W3C schemas are very hard to implement, while Relax NG are basically automata.
- DocBook, OpenOffice, SVG, or XHTML use Relax NG.
- Relax NG can use the best of W3C schemas, i.e. the simple types.
- ISO Schematron handles best non-structural constraints.
- In general, for W3C XML Schemas vs. Relax NG read
  `http://www.imc.org/ietf-xml-use/mail-archive/msg00217.html`

Layering

- One PML schema per annotation layer.
- The annotation layers are interconnected by PML links.
- If the annotation structure allows it, the `KNIT` role can be used to allow for merging annotation layers.

For example:

- PDT 2.0 uses the following PML schemas: `wdata_schema.xml`, `mdata_schema.xml`, `tdata_schema.xml`, and `adata_schema.xml`.
- via `KNIT`, w-layer can be merged into m-layer, m-layer can be merged into a-layer.



Intermediate layers

- Different machine processing strategies may have different views on what compounds a single layer.

  A typical example is tokenization and sentence-break segmentation (usually done before tagging, but for some languages, such as Arabic, it may be reasonable to do all at the same time).
- Intermediate layers can also result from incomplete manual annotation as fragments of the final annotation layers.

  PML schemas for annotation formats can be derived by PML modularization support.

PML representation of PDT 2.0 dependency layers

a-layer instance consists of:
- meta data (annotation info)
- a list of trees (each by a technical root node structure)

Two types of nodes (both structures)
- technical root (a-root.type)
- analytical node (a-node.type)

Each node carries:
- member `ord` providing tree ordering (role `#ORDER`)
- a list of child nodes (member `children` with role `#CHILDNODES`)
- pointers to m-layer (with role `#KNIT` on a-node.type)
- analytical function `afun` (const. `AuxS` on root, enumerated elsewhere)
- coordination/apposition and parenthesis membership flags

t-layer layer follows the same pattern for representing ordered dependency trees, only t-node structures carry much richer annotation and some extra relational stuff, like co-reference links and quotation sets.

PML schema for a-layer (1)

```
<pml_schema
    xmlns="http://ufal.mff.cuni.cz/pdt/pml/schema/" version="1.1">
  <revision>1.0.3</revision>
  <description>PDT 2.0 analytical trees</description>
  <reference name="mdata" readas="dom"/>
  <reference name="wdata" readas="dom"/>

  <import schema="mdata_schema.xml" type="m-node.type"
    minimal_revision="1.0.3"/>
  <import schema="mdata_schema.xml" type="bool.type"/>

  <derive type="m-node.type">
    <structure name="m-node">
      <member name="id" as_attribute="1" role="#ID" required="1">
        <cdata format="PMLREF"/>
      </member>
    </structure>
  </derive>
...
```

PML schema for a-layer (2)

```
...
<root name="adata" type="a-adata.type"/>

<type name="a-adata.type">
  <structure>
    <member name="meta" required="0" type="a-meta.type"/>
    <member name="trees" role="#TREES" required="1">
      <list type="a-root.type" ordered="1"/>
    </member>
  </structure>
</type>

<type name="a-meta.type">
  <structure>
    <member name="annotation_info">
      <structure name="a-annotation-info">
        <member name="version_info"><cdata format="any"/></member>
        <member name="desc"><cdata format="any"/></member>
      </structure>
    </member>
  </structure>
</type>
...
```

PML schema for a-layer (3)

```
...
<type name="a-root.type">
  <structure role="#NODE" name="a-root">
    <member name="id" role="#ID" as_attribute="1" required="1">
      <cdata format="ID"/>
    </member>
    <member name="s.rf"><cdata format="PMLREF"/></member>
    <member name="afun"><constant>AuxS</constant></member>
    <member name="ord" role="#ORDER" required="1">
      <cdata format="nonNegativeInteger"/>
    </member>
    <member name="children" role="#CHILDNODES">
      <list type="a-node.type" ordered="1"/>
    </member>
  </structure>
</type>
...
```

PML schema for a-layer (4)

```
...
<type name="a-node.type">
  <structure role="#NODE" name="a-node">
    <member name="id" role="#ID" as_attribute="1" required="1">
      <cdata format="ID"/>
    </member>
    <member name="m.rf" role="#KNIT" type="m-node.type">
      <cdata format="PMLREF"/>
    </member>
    <member name="afun" type="a-afun.type" required="1"/>
    <member name="is_member" type="bool.type"/>
    <member type="bool.type" name="is_parenthesis_root"/>
    <member name="ord" role="#ORDER" required="1">
      <cdata format="nonNegativeInteger"/>
    </member>
    <member name="children" role="#CHILDNODES">
      <list type="a-node.type" ordered="1"/>
    </member>
  </structure>
</type>
...
```

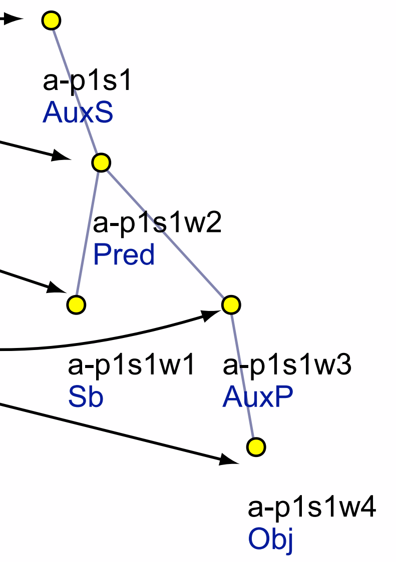
Sample a-layer instance

```
<adata xmlns="http://ufal.mff.cuni.cz/pdt/pml/">
 <head>
  <schema href="adata_schema.xml" />
  <references>
   <reffile id="m" name="mdata" href="sample4.m.gz" />
   <reffile id="w" name="wdata" href="sample4.w.gz" />
  </references>
 </head>
 <meta>
  <annotation_info>
   <desc>Manual annotation</desc>
  </annotation_info>
 </meta>
 <trees>
  <LM id="a-p1s1">...</LM>
  <LM id="a-p1s2">...</LM>
  ...
 </trees>
```

Sample PDT 2.0 instance (a-layer)

a-layer

```
<LM id="a-p1s1">
<s.rf>m#m-p1s1</s.rf>
<ord>0</ord>
<children>
  <LM id="a-p1s1w2">
    <m.rf>m#m-p1s1w2</m.rf>
    <afun>Pred</afun>
    <ord>2</ord>
    <children>
      <LM id="a-p1s1w1">
        <m.rf>m#m-p1s1w1</m.rf>
        <afun>Sb</afun>
        <ord>1</ord>
      </LM>
      <LM id="a-p1s1w3">
        <m.rf>m#m-p1s1w3</m.rf>
        <afun>AuxP</afun>
        <ord>3</ord>
        <children id="a-p1s1w4">
          <m.rf>m#m-p1s1w4</m.rf>
          <afun>Obj</afun>
          <ord>4</ord>
        </children>
      </LM>
    </children>
  </LM>
</children>
</LM>
```



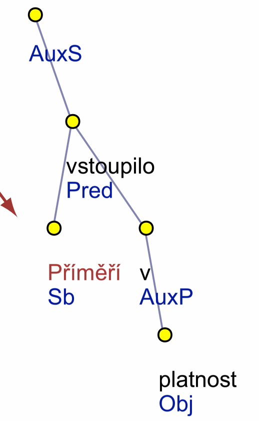
Sample PDT 2.0 instance (a+m-layer)

a-layer

```
<LM id="a-p1s1">
<s.rf>m#m-p1s1</s.rf>
<ord>0</ord>
<children>
  <LM id="a-p1s1w2">
    <m.rf>m#m-p1s1w2</m.rf>
    <afun>Pred</afun>
    <ord>2</ord>
    <children>
      <LM id="a-p1s1w1">
        <m.rf>m#m-p1s1w1</m.rf>
        <afun>Sb</afun>
        <ord>1</ord>
      </LM>
      <LM id="a-p1s1w3">
        <m.rf>m#m-p1s1w3</m.rf>
        <afun>AuxP</afun>
        <ord>3</ord>
        <children id="a-p1s1w4">
          <m.rf>m#m-p1s1w4</m.rf>
          <afun>Obj</afun>
          <ord>4</ord>
        </children>
      </LM>
    </children>
  </LM>
</children>
</LM>
```

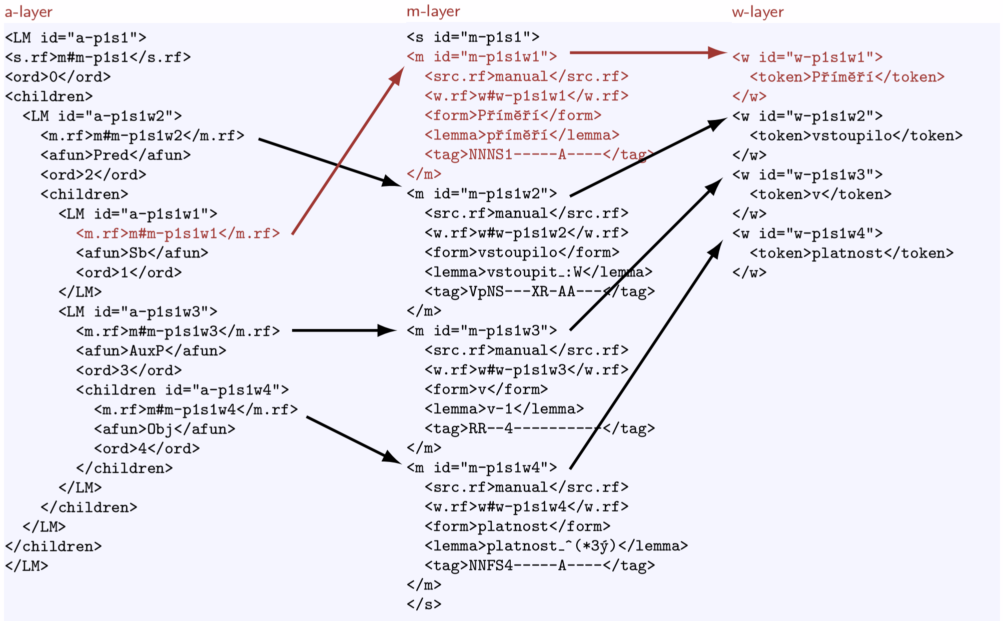
m-layer

```
<s id="m-p1s1">
<m id="m-p1s1w1">
  <src.rf>manual</src.rf>
  <w.rf>w#w-p1s1w1</w.rf>
  <form>Příměří</form>
  <lemma>příměří</lemma>
  <tag>NNNS1-----A----</tag>
</m>
<m id="m-p1s1w2">
  <src.rf>manual</src.rf>
  <w.rf>w#w-p1s1w2</w.rf>
  <form>vstoupilo</form>
  <lemma>vstoupit_:W</lemma>
  <tag>VpNS---XR-AA---</tag>
</m>
<m id="m-p1s1w3">
  <src.rf>manual</src.rf>
  <w.rf>w#w-p1s1w3</w.rf>
  <form>v</form>
  <lemma>v-1</lemma>
  <tag>RR--4----------</tag>
</m>
<m id="m-p1s1w4">
  <src.rf>manual</src.rf>
  <w.rf>w#w-p1s1w4</w.rf>
  <form>platnost</form>
  <lemma>platnost_^(*3ý)</lemma>
  <tag>NNFS4-----A----</tag>
</m>
</s>
```



Příměří vstoupilo v platnost

Sample PDT 2.0 instance (a+m+w-layer)



a-layer

```
<LM id="a-p1s1">
<s.rf>m#m-p1s1</s.rf>
<ord>0</ord>
<children>
  <LM id="a-p1s1w2">
    <m.rf>m#m-p1s1w2</m.rf>
    <afun>Pred</afun>
    <ord>2</ord>
    <children>
      <LM id="a-p1s1w1">
        <m.rf>m#m-p1s1w1</m.rf>
        <afun>Sb</afun>
        <ord>1</ord>
      </LM>
      <LM id="a-p1s1w3">
        <m.rf>m#m-p1s1w3</m.rf>
        <afun>AuxP</afun>
        <ord>3</ord>
        <children id="a-p1s1w4">
          <m.rf>m#m-p1s1w4</m.rf>
          <afun>Obj</afun>
          <ord>4</ord>
        </children>
      </LM>
    </children>
  </LM>
</children>
</LM>
```

m-layer

```
<s id="m-p1s1">
<m id="m-p1s1w1">
  <src.rf>manual</src.rf>
  <w.rf>w#w-p1s1w1</w.rf>
  <form>Příměří</form>
  <lemma>příměří</lemma>
  <tag>NNNS1-----A----</tag>
</m>
<m id="m-p1s1w2">
  <src.rf>manual</src.rf>
  <w.rf>w#w-p1s1w2</w.rf>
  <form>vstoupilo</form>
  <lemma>vstoupit_:W</lemma>
  <tag>VpNS---XR-AA---</tag>
</m>
<m id="m-p1s1w3">
  <src.rf>manual</src.rf>
  <w.rf>w#w-p1s1w3</w.rf>
  <form>v</form>
  <lemma>v-1</lemma>
  <tag>RR--4----------</tag>
</m>
<m id="m-p1s1w4">
  <src.rf>manual</src.rf>
  <w.rf>w#w-p1s1w4</w.rf>
  <form>platnost</form>
  <lemma>platnost_^(*3ý)</lemma>
  <tag>NNFS4-----A----</tag>
</m>
</s>
```

w-layer

```
<w id="w-p1s1w1">
  <token>Příměří</token>
</w>
<w id="w-p1s1w2">
  <token>vstoupilo</token>
</w>
<w id="w-p1s1w3">
  <token>v</token>
</w>
<w id="w-p1s1w4">
  <token>platnost</token>
</w>
```

Sample PDT 2.0 instance (a+m+w-layers knitted)

a-layer

```
<LM id="a-p1s1">
<s.rf>m#m-p1s1</s.rf>
<ord>0</ord>
<children>
  <LM id="a-p1s1w2">
    <m.rf>m#m-p1s1w2</m.rf>
    <afun>Pred</afun>
    <ord>2</ord>
    <children>
      <LM id="a-p1s1w1">
        <m.rf>m#m-p1s1w1</m.rf>
        <afun>Sb</afun>
        <ord>1</ord>
      </LM>
      <LM id="a-p1s1w3">
        <m.rf>m#m-p1s1w3</m.rf>
        <afun>AuxP</afun>
        <ord>3</ord>
        <children id="a-p1s1w4">
          <m.rf>m#m-p1s1w4</m.rf>
          <afun>Obj</afun>
          <ord>4</ord>
        </children>
      </LM>
    </children>
  </LM>
</children>
</LM>
```

m-layer

```
<s id="m-p1s1">
```

w-layer

After knitting

```
<LM id="a-p1s1w1">
  <m id="m#m-p1s1w1">
    <src.rf>manual</src.rf>
    <w id="w#w-p1s1w1">
      <token>Příměří</token>
    </w>
    <form>Příměří</form>
    <lemma>příměří</lemma>
    <tag>NNNS1-----A----</tag>
  </m>
  <afun>Sb</afun>
  <ord>1</ord>
</LM>
```

```
</m>
</s>
```

Technical issues with multi-layered annotations in PML

In PDT 2.0, the full annotation of each document comprises of four PML instances (one per layer), which contain references to one another.

This raises some technical obstacles to the users and tool implementators:

- PML instances are not copy-safe (one cannot simply rename e.g. the m-layer instance without fixing the reference to it in the a-layer instance)
- PDT 2.0 data are distributed in gzip-compressed forms. Same problem: decompressing removes the `.gz` suffix, which renames the instance. Again, references have to be fixed.

Future development

- foreign namespaces

  allow XML data from non-PML namespaces within PML instance (MathML, XLink, RDF,...)
- meta-data

  uniform representation of meta-data (via RDF)
- schema annotation

  similar to what W3C schema has
- new roles

  Current set of roles doesn't cover all situations, e.g. lexicons
- guidelines for more types of links

  foreign XML, other media (text, audio, video, graphics, . . .)
- automatic data-binding and API generation

  translate PML schemas into a ready-to-use library with optimal in-memory representation, validation, parser, serializer, indexing, etc.

PML homepage

`http://ufal.mff.cuni.cz/jazz/PML`

- PML spec (with many examples)
- Relax NG schema for PML schema
- tools (simplification, validation, ...)
- updated PDT 2.0 schemas
- links

Part 7

TOOLS (continued): Automatic processing of data

STYX
- an electronic exercise book of Czech

Prague Treebanking for Everyone

Automatic Processing of Data

Jan Štěpánek

29th November 2006

`run_all` from **PDT**

Located in `tools/machine-annotation/run_all`.

- Tokenization of the input plain text and segmentation into sentences.
- Morphological analysis and tagging (morphological disambiguation).
- Dependency parsing.
- Analytical (dependency) function assignment for all nodes of the parsed tree.

Limitations and requirements:

- Written in **C/C++**, **perl** and **tcsh**.
- Compiled for Linux on an i386 architecture.

`run_all` from **PDT**

Tokenization and segmentation

- Problems with full-stop (“.”) in Czech.
- Tested on `amw` data:
  - Segmentation:
    precision 98.0 %, recall 91.4 %, F-measure 94.6 %.
  - Tokenization:
    precision 100.0 %, recall 99.2 %, F-measure 99.6 %.

`run_all` from **PDT**

Morphological analysis

- All possible lemmas and tags.
- Dictionary of 350,000 entries, 12 million Czech word forms.
- Error rate: 2.5 % (foreign names and typos).

`run_all` from **PDT**
Morphological tagger

- Maximum entropy approach with greedy incorporation of selectors.
- Tagging – 93.08 % accuracy on evaluation test data.

`run_all` from **PDT**
Parsing

- Czech adaptation of the parser of Michael Collins — dependency based.
- Only shorter sentences (up to 60 words).
- Evaluation test data: 81.6 % parents assigned correctly (both training and test data tagged machinely).

`run_all` from **PDT**
Analytical function assignment

- Decision tree approach (Quinlan's **C5** classifier translated to **perl**)
- Uses **btred**.
- Precision around 92 %.

`run_all` from **PDT**
Conversion to **PML**

- All the previous steps use deprecated **CSTS** format.
- Conversion script uses **btred**.

`morph_chain` from **CAC**

Located in `tools/morph_chain`.

- Hidden Markov models — trained by Viterbi algorithm + averaged perceptron for evaluating transitions between HMM states (Collins)
- Trained on **PDT**: 91.8 % (93.1)

Dealing with the Structural Annotation

btred

Features (same as those of **TrEd**):

- Independent on operating system (MS Windows, Linux, OS X...)
- Open source program, available for free
- Written in **perl** (macros, predefined functions)

Requirements and installation:

- **perl** (5.8.3 or newer) with **Tk** library
- `http://ufal.mff.cuni.cz/~pajas/tred/`
  - For MS Windows: `tred_wininst_en.zip` → `setup.bat`
  - For Linux: `tred-dep-unix.tar.gz` → `install tred-current.tar.gz`

Dealing with the Structural Annotation

Why **btred**?

- Object-oriented tree representation — a rich repertory of basic functions for tree traversing and for many other basic operations on trees + several highly non-trivial functions suitable for linguistically motivated traversing of trees (e.g. solving the coordination relations).
- Reasonable stability because of long-time experience (development of **PDT**).
- Network (parallel) version (not for MS Windows).
- Powerful and fast search-engine (pipes).

Dealing with the Structural Annotation

Simplest examples

Basic syntax:

`btred -e` <*code*> `file(s)` OR `btred -I macro_file file(s)`

```
$ btred -e 'writeln("Hello world!");' sample0.a.gz
BTRED: Trying /export/common/lib/tred
Config file: /home/stepanek/.tredrc
BTRED: Resource path: /home/stepanek/tred/resources/
BTRED: Reading macros from /usr/tred/tred.mac...
BTRED: done.
BTRED: <script>
package TredMacro;
sub _btred_eval_ {
  writeln("Hello world!");
}
;
;
</script>
BTRED: Processing: sample0.a.gz (1/1)
Hello world!
BTRED: Done.
```

Dealing with the Structural Annotation
Simplest examples (2)

Traversing trees:

```
$ btred -QTe ’writeln($a++);’ sample0.a.gz
 ...
52
53
```

Traversing trees *and nodes*:

```
$ btred -QNTe ’writeln($a++);’ sample0.a.gz
 ...
864
865
```

More files:

```
$ btred -QNTe ’writeln($a++);’ sample*.a.gz
 ...
7813
7814
```

Dealing with the Structural Annotation
Simple examples

Simple attributes:

```
$ btred -QNTe ’writeln($this->{afun})’ sample0.a.gz
 ...
Atr
AuxK
```

Structured and list attributes

```
$ btred -QNTe ’writeln($this->attr("m/form"))’ sample0.a.gz
 ...
založení
OSN
.
$ btred -QNTe ’my @ids = ListV($this->attr("coref_text.rf"));
  if (@ids){
    writeln(PML_T::GetNodeByID($ids[0])->{t_lemma});
  }’ sample*.t.gz
 ...
#PersPron
Chodura
```

Dealing with the Structural Annotation
Examples

Methods — find the tree with the highest number of nodes (root `descendants`):

```
$ btred -QTe ’writeln(scalar($root->descendants))’
   sample*.t.gz | sort -n | tail -n1
 42
```

Similarly: `children`, `parent`, `lbrother`…

Dealing with the Structural Annotation
Examples (2)

perl functions `grep` and `map` — print verbs and their objects:

```
$ btred -QNTe ’if($this->attr("m/tag") =~ /^V/ ) {
  writeln join " ",
    $this->attr("m/form"),
    map {$_->attr("m/form")}
      grep {$_->{afun} eq "Obj"} $this->children;
}’ sample1.a.gz
 ...
Nehodlá vyjadřovat
vyjadřovat
dokončil šetření
předal spis zastupitelství
```

Similarly: `first`

Dealing with the Structural Annotation
Complex examples

Effective children and parents — what semantical part of speech are the parents of **actors** and how often:

```
$ btred -QNTe '
  my $par;
  $par = join(" ",
    map( {
      $_->attr("gram/sempos")
    } PML_T::GetEParents())
  ),writeln($par) if $this->{functor} eq "ACT"
' sample*.t.gz | sort | uniq -c | sort -n
 ...
4 v v v
5 adj.denot
25
30 v v
108 n.denot
117 n.denot.neg
667 v
```

Dealing with the Structural Annotation
Complex examples (2)

Crossing layer boundaries — count all **actors** expressed by a noun in nominative (1^{st} case):

```
$ btred -QNTe 'writeln() if $this->{functor} eq "ACT"
  and ! $this->{is_generated}
  and first {
    my $t = $_->attr("m/tag");
    $t =~ /^N...1/
  } PML_T::GetANodes($this)
' sample*.t.gz | wc -l
 422
```

Dealing with the Structural Annotation
Searching and viewing results

TrEd function `FPosition()`:

```
$ btred -QNTe 'FPosition()
  if $this->{t_lemma} =~ /_.*_/' sample*.t.gz
 sample9.t.gz##14.22
```

```
$ btred -I macro-that-uses-FPosition *.t.gz |
    tred -l-
```

Dealing with the Structural Annotation
Speeding up

Crawling through all the tectogrammatical nodes by **btred** takes about *10 minutes*. Most time is spent by *opening and parsing* the data.

Possible solution: read all the data just once and keep them in the memory.

Problem: not enough memory.

Solution: distribute the data among several computers.

ntred (network-tred): **btred** servers + *hub*

Dealing with the Structural Annotation
Speeding up (2)

ntred requirements:

- Cannot run on MS Windows (problems with net sockets).
- All the computers running **btred**-servers must share a filesystem.
- Password-free access to all the computers is needed.
- Some macros have to be adjusted (e.g. overall statistics).

STYX

Prague Dependency Treebank as an exercise book of Czech

Ondřej Kučera

Contents

1. **Introduction** (motivation, PDT, implementation)
2. **Filtering sentences**
3. **Transformations of trees**
4. **STYX:** FilterSentences, Charon, Styx

Motivation

- children of today use computers regularly
  - games, web surfing, chatting, writing, drawing
- why couldn't they parse sentences or determine parts of speech?

Building an exercise book

Manually

- extremely hard
  - choose (make up) the sentences
  - annotate them
- considerably limited number of sentences
- often too simple sentences not reflecting the real usage of the language

Building an exercise book

Automatically

- if we have annotated data
- the work of choosing the sentences and annotating them is already done
- the data in corpus reflect the real usage of the language
- the number of sentences corresponds to the size of the corpus
- PDT

Prague Dependency Treebank

- annotated on four layers (word, morphological, analytical, tectogrammatical)
- inner data format: PML (Prague Markup Language) – based on XML

PDT vs. school syntax

- annotation rules of PDT allow to process any sentence
  ⇨ filtering sentences
- Analytical layer of PDT differs from the school syntax in many ways
  ⇨ transformations of analytical trees

Filtering sentences

Filtering in numbers

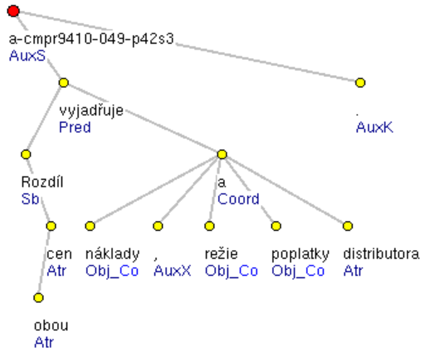
- nine different filters
- starting number of sentences: 49,442
- after application of the filters: 11,705
- about 23.7% sentences kept

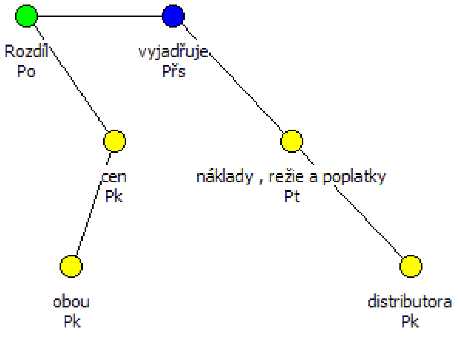
Transformation of trees

- three basic transformations
- particular transformations consist of
  - combining of the three basic transformations
  - rules for modification of the syntactic functions

Transformations of trees

Example





Implementation

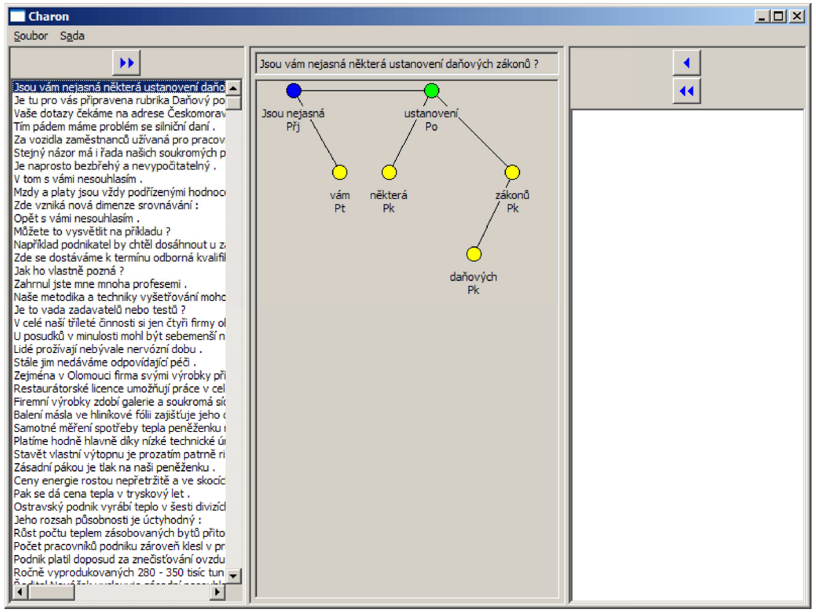
Java

- high-level language with number of mechanisms protecting programmers “against themselves”
- portability
- presence of SWT library

SWT

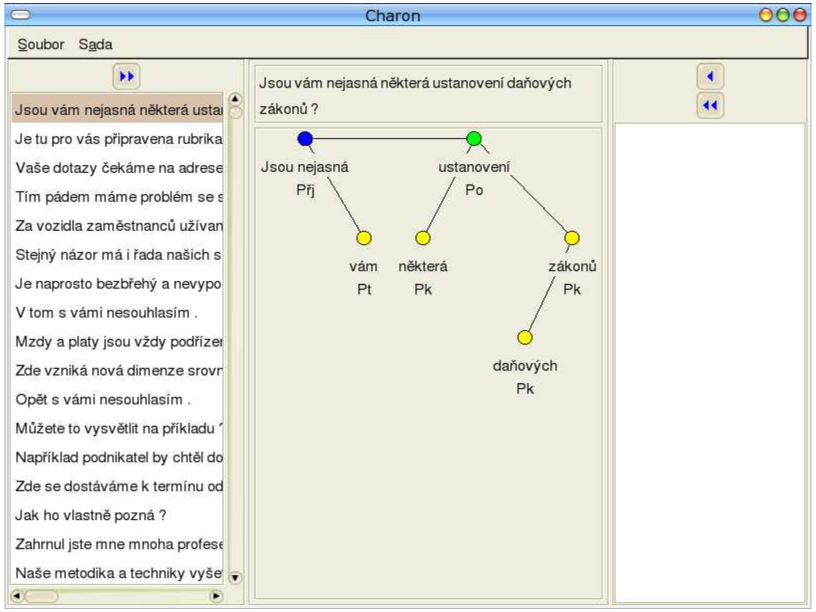
- Standard Widget Toolkit
- provides native look and feel of graphical user interface
- speed

Implementation



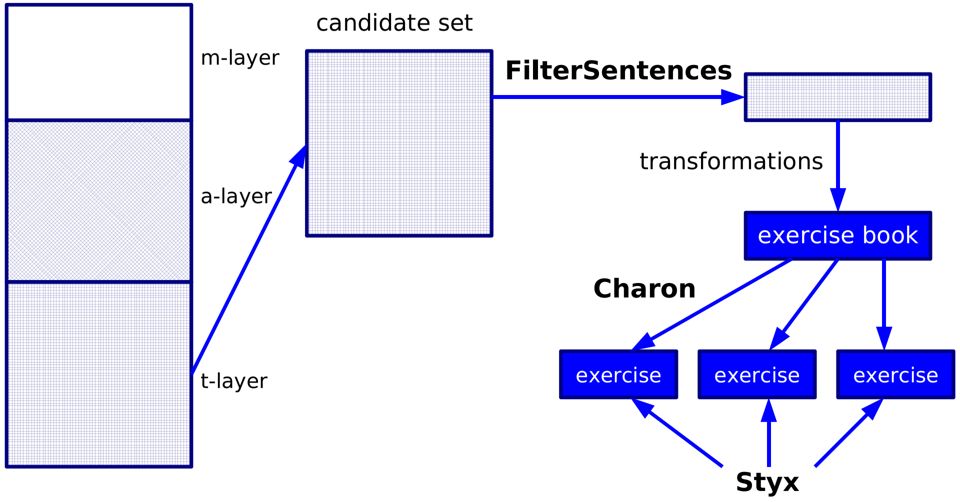
Jsou vám nejasná některá ustanovení daňo
Je tu pro vás připravena rubrika Daňový po
Vaše dotazy čekáme na adrese Českomorav
Tím pádem máme problém se silniční daní .
Za vozidla zaměstnanců užívaná pro pracov
Stejný názor má i řada našich soukromých p
Je naprosto bezbřehý a nevypočitatelný .
V tom s vámi nesouhlasím .
Mzdy a platy jsou vždy podřízenými hodnoc
Zde vzniká nová dimenze srovnávání :
Opět s vámi nesouhlasím .
Můžete to vysvětlit na příkladu ?
Například podnikatel by chtěl dosáhnout u z
Zde se dostáváme k termínu odborná kvalifi
Jak ho vlastně pozná ?
Zahrnul jste mne mnoha profesemi .
Naše metodika a techniky vyšetřování mohc
Je to vada zadavatelů nebo testů ?
V celé naší tříleté činnosti si jen čtyři firmy ol
U posudků v minulosti mohl být sebemenší n
Lidé prožívají nebývale nervózní dobu .
Stále jim nedáváme odpovídající péči .
Zejména v Olomouci firma svými výrobky při
Restaurátorské licence umožňují práce v cel
Firemní výrobky zdobí galerie a soukromá síc
Balení másla ve hliníkové fólii zajišťuje jeho c
Samotné měření spotřeby tepla peněženku r
Platíme hodně hlavně díky nízké technické ú
Stavět vlastní výtopnu je prozatím patrně ri
Zásadní pákou je tlak na naši peněženku .
Ceny energie rostou nepřetržitě a ve skocíc
Pak se dá cena tepla v tryskový let .
Ostravský podnik vyrábí teplo v šesti divizíc
Jeho rozsah působnosti je úctyhodný :
Růst počtu teplem zásobovaných bytů přito
Počet pracovníků podniku zároveň klesl v pr
Podnik platil doposud za znečisťování ovzdu
Ročně vyprodukovaných 280 - 350 tisíc tun

Implementation



Jsou vám nejasná některá usta
Je tu pro vás připravena rubrika
Vaše dotazy čekáme na adrese
Tím pádem máme problém se s
Za vozidla zaměstnanců užívan
Stejný názor má i řada našich s
Je naprosto bezbřehý a nevypo
V tom s vámi nesouhlasím .
Mzdy a platy jsou vždy podřízer
Zde vzniká nová dimenze srovr
Opět s vámi nesouhlasím .
Můžete to vysvětlit na příkladu
Například podnikatel by chtěl do
Zde se dostáváme k termínu od
Jak ho vlastně pozná ?
Zahrnul jste mne mnoha profese
Naše metodika a techniky vyše

STYX:
FilterSentences, Charon, Styx



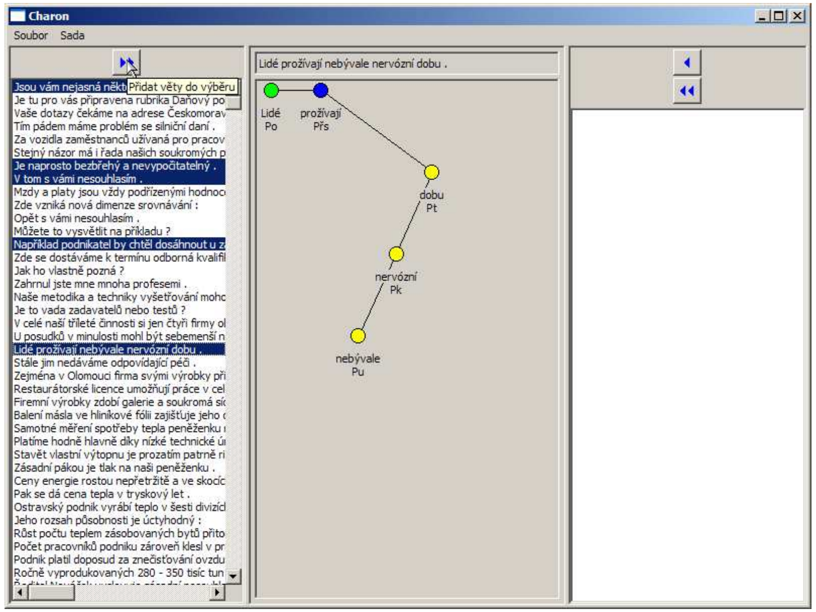
FilterSentences

- used for applying the filters
- reads data in PML format
- each sentence is tested by a filter
- output data contains the sentences that the filter kept
- output again in PML format

Charon

- “administrative” program
- loads all sentences available
- the user selects sentences that he or she wants to have in the exercise
- in the end the user saves the exercise

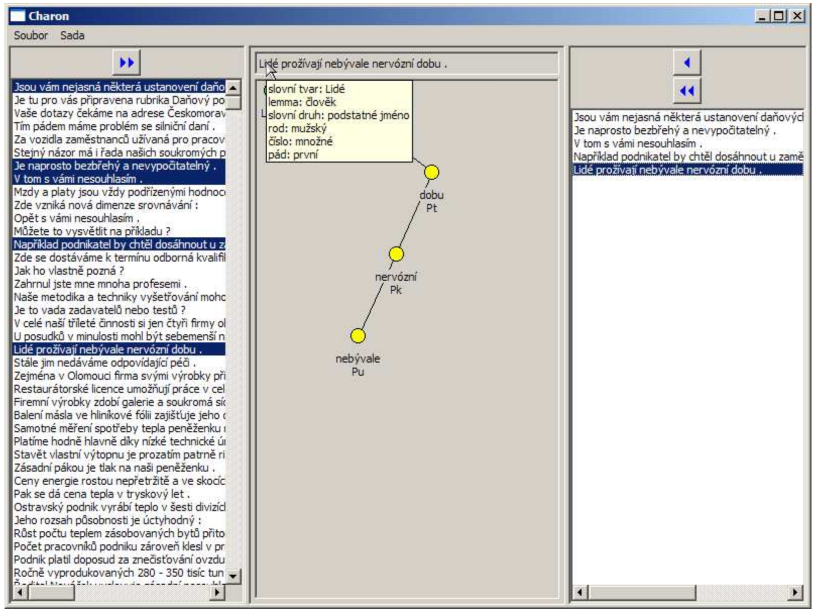
Charon



Jsou vám nejasná někt
Je tu pro vás připravena rubrika Daňový po
Vaše dotazy čekáme na adrese Českomorav
Tím pádem máme problém se silniční daní .
Za vozidla zaměstnanců užívaná pro pracov
Stejný názor má i řada našich soukromých p
Je naprosto bezbřehý a nevypočitatelný .
V tom s vámi nesouhlasím .
Mzdy a platy jsou vždy podřízenými hodnoc
Zde vzniká nová dimenze srovnávání :
Opět s vámi nesouhlasím .
Můžete to vysvětlit na příkladu ?
Například podnikatel by chtěl dosáhnout u z
Zde se dostáváme k termínu odborná kvalifi
Jak ho vlastně pozná ?
Zahrnul jste mne mnoha profesemi .
Naše metodika a techniky vyšetřování mohc
Je to vada zadavatelů nebo testů ?
V celé naší tříleté činnosti si jen čtyři firmy ol
U posudků v minulosti mohl být sebemenší n
Lidé prožívají nebývale nervózní dobu .
Stále jim nedáváme odpovídající péči .
Zejména v Olomouci firma svými výrobky při
Restaurátorské licence umožňují práce v cel
Firemní výrobky zdobí galerie a soukromá sí
Balení másla ve hliníkové fólii zajišťuje jeho
Samotné měření spotřeby tepla peněženku
Platíme hodně hlavně díky nízké technické ú
Stavět vlastní výtopnu je prozatím patrně ri
Zásadní pákou je tlak na naši peněženku .
Ceny energie rostou nepřetržitě a ve skoků
Pak se dá cena tepla v tryskový let .
Ostravský podnik vyrábí teplo v šesti divizíc
Jeho rozsah působnosti je úctyhodný :
Růst počtu teplem zásobovaných bytů přito
Počet pracovníků podniku zároveň klesl v pr
Podnik platil doposud za znečisťování ovzdu
Ročně vyprodukovaných 280 - 350 tisíc tun

Lidé prožívají nebývale nervózní dobu .

Charon



Jsou vám nejasná některá ustanovení daňo
Je tu pro vás připravena rubrika Daňový po
Vaše dotazy čekáme na adrese Českomorav
Tím pádem máme problém se silniční daní .
Za vozidla zaměstnanců užívaná pro pracov
Stejný názor má i řada našich soukromých p
Je naprosto bezbřehý a nevypočitatelný .
V tom s vámi nesouhlasím .
Mzdy a platy jsou vždy podřízenými hodnoc
Zde vzniká nová dimenze srovnávání :
Opět s vámi nesouhlasím .
Můžete to vysvětlit na příkladu ?
Například podnikatel by chtěl dosáhnout u z
Zde se dostáváme k termínu odborná kvalifi
Jak ho vlastně pozná ?
Zahrnul jste mne mnoha profesemi .
Naše metodika a techniky vyšetřování mohc
Je to vada zadavatelů nebo testů ?
V celé naší tříleté činnosti si jen čtyři firmy ol
U posudků v minulosti mohl být sebemenší n
Lidé prožívají nebývale nervózní dobu .
Stále jim nedáváme odpovídající péči .
Zejména v Olomouci firma svými výrobky při
Restaurátorské licence umožňují práce v cel
Firemní výrobky zdobí galerie a soukromá sí
Balení másla ve hliníkové fólii zajišťuje jeho
Samotné měření spotřeby tepla peněženku
Platíme hodně hlavně díky nízké technické ú
Stavět vlastní výtopnu je prozatím patrně ri
Zásadní pákou je tlak na naši peněženku .
Ceny energie rostou nepřetržitě a ve skoků
Pak se dá cena tepla v tryskový let .
Ostravský podnik vyrábí teplo v šesti divizíc
Jeho rozsah působnosti je úctyhodný :
Růst počtu teplem zásobovaných bytů přito
Počet pracovníků podniku zároveň klesl v pr
Podnik platil doposud za znečisťování ovzdu
Ročně vyprodukovaných 280 - 350 tisíc tun

Lidé prožívají nebývale nervózní dobu .

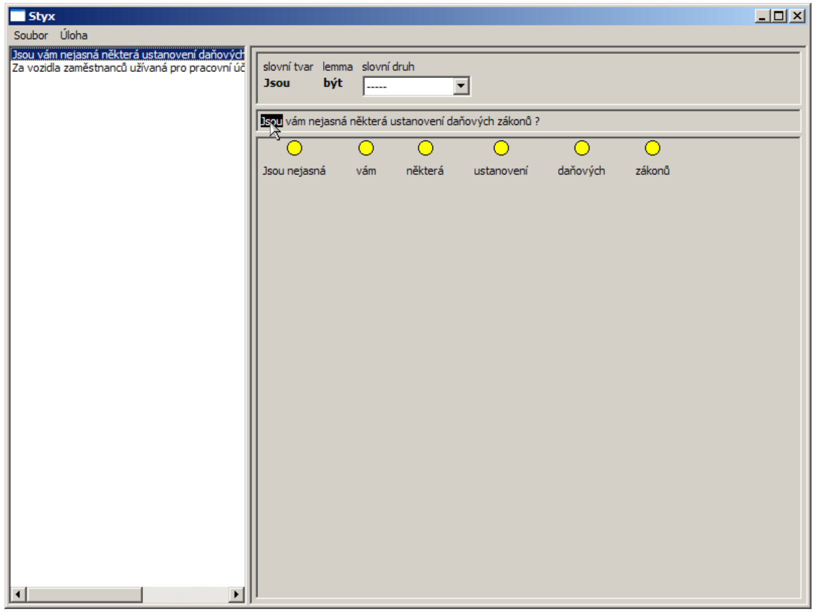
slovní tvar: Lidé
lemma: člověk
slovní druh: podstatné jméno
rod: mužský
číslo: množné
pád: první

Jsou vám nejasná některá ustanovení daňových
Je naprosto bezbřehý a nevypočitatelný .
V tom s vámi nesouhlasím .
Například podnikatel by chtěl dosáhnout u zamě
Lidé prožívají nebývale nervózní dobu .

Styx

- exercise book itself
- user loads an exercise previously created and saved in Charon

Styx



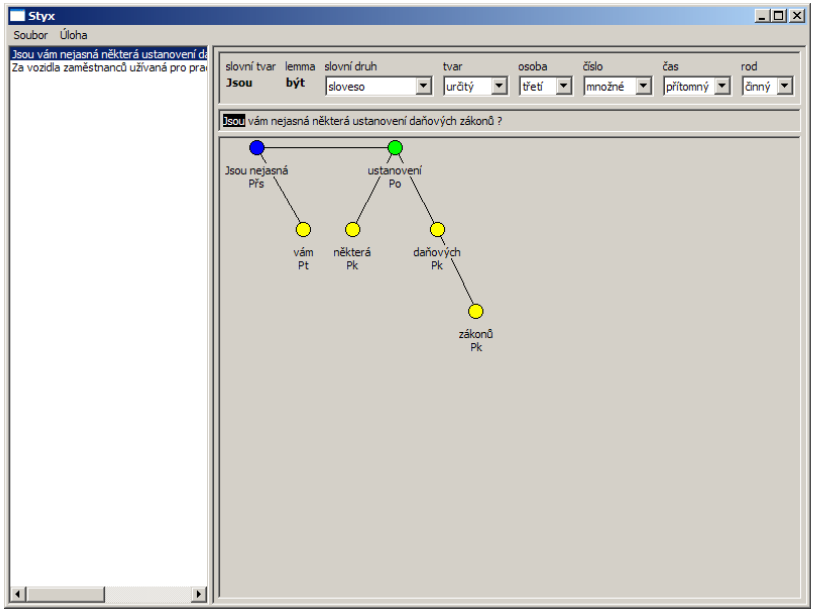
Jsou vám nejasná některá ustanovení daňových

Za vozidla zaměstnanců užívaná pro pracovní úč

slovní tvar: **Jsou**
lemma: **být**
slovní druh: -----

Jsou vám nejasná některá ustanovení daňových zákonů ?

Styx



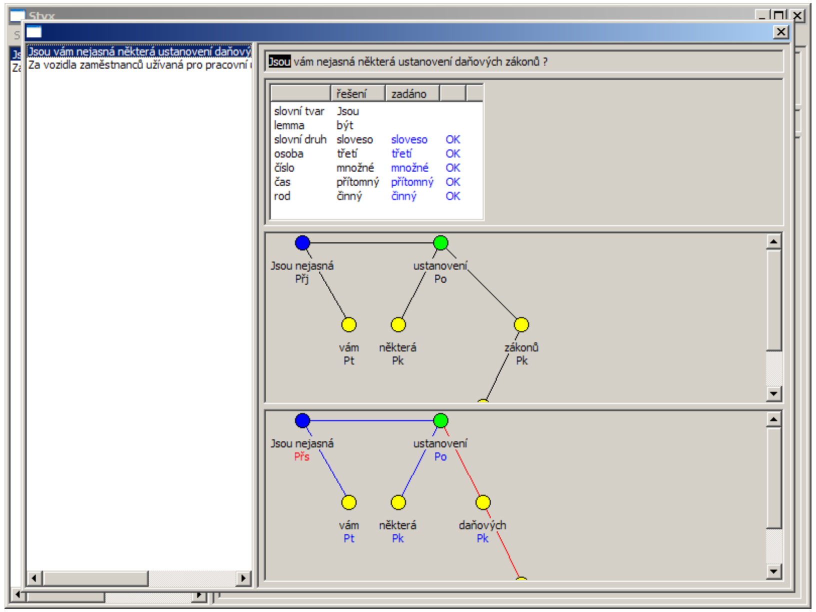
Jsou vám nejasná některá ustanovení da

Za vozidla zaměstnanců užívaná pro pra

slovní tvar: **Jsou**
lemma: **být**
slovní druh: sloveso
tvar: určitý
osoba: třetí
číslo: množné
čas: přítomný
rod: činný

Jsou vám nejasná některá ustanovení daňových zákonů ?

Styx



Jsou vám nejasná některá ustanovení daňový

Za vozidla zaměstnanců užívaná pro pracovní

Jsou vám nejasná některá ustanovení daňových zákonů ?

	řešení	zadáno	
slovní tvar	Jsou		
lemma	být		
slovní druh	sloveso	sloveso	OK
osoba	třetí	třetí	OK
číslo	množné	množné	OK
čas	přítomný	přítomný	OK
rod	činný	činný	OK

Questions

and perhaps some answers…

Part 8

DATA:
More Prague Treebanks

The Prague Czech-English Dependency Treebank (part 8.1)

Jan Hajič
Institute of Formal and Applied Linguistics
School of Computer Science
Faculty of Mathematics and Physics
Charles University, Prague
Czech Republic

The Goal:
Parallel, Annotated Treebank

- Parallel corpora
  - Comparative/contrastive and translation studies
  - Semantics
  - Other “linguistic research goals”
- Machine Translation
  - “Training” material
    - Human-translated texts
  - Testing material
    - Evaluation – human, automatic

The PCEDT

- One of “family” of PDT-like treebanks
- Texts:
  - Wall Street Portion of the Penn Treebank, ver. III
  - Czech translation (manual) of the above
- Size
  - 1.2 million words, ~50,000 sentences
- Annotation
  - All 4 layers as in PDT: tokens, morphology, syntax, tectogrammatical representation

---

Penn Treebank

- University of Pennsylvania, 1993
  - Linguistic Data Consortium
- Wall Street Journal texts
  - 1989-1991
  - Financial (most), news, arts, sports
  - 2499 documents in 25 sections
- Annotation
  - POS (Part-of-speech tags)
  - Syntactic “bracketing” + bracket (syntactic) labels
  - (Syntactic) Function tags

Penn Treebank Example

- ((S
- (NP-SBJ
- (NP (NNP Pierre) (NNP Vinken))
- (, ,)
- (ADJP
- (NP (CD 61) (NNS years))
- (JJ old))
- (, ,))
- (VP (MD will)
- (VP (VB join)
- (NP (DT the) (NN board))
- (PP-CLR (IN as)
- (NP (DT a) (JJ nonexecutive) (NN director)))
- (NP-TMP (NNP Nov.) (CD 29))))
- (. .)))

"Preterminal"
POS tag (NNS)
(noun, plural)

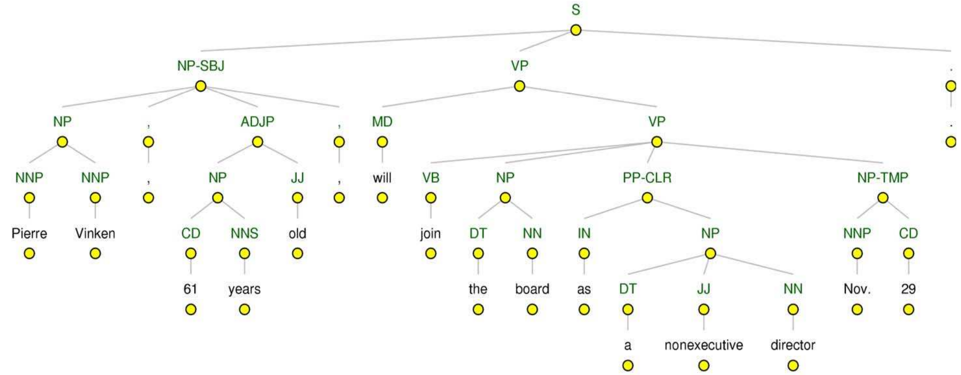
Noun Phrase

Phrase label (NP)

Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.

Penn Treebank Example: Sentence Tree

- Phrase-based tree representation:



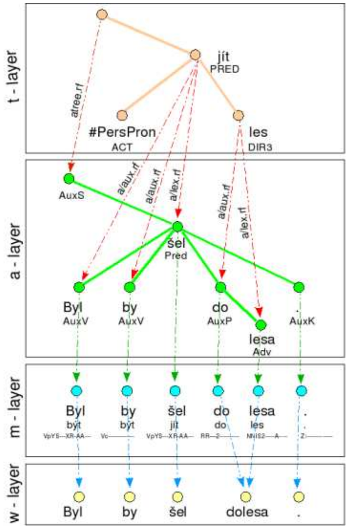
PDT Layers of Annotation

Tectogrammatical structure

Surface syntax

Morphology

Tokens (words)



Parallel Czech-English Annotation

- English text -> Czech text (human translation)
- Czech side (goal): all layers manual annotation
- English side (goal):
  - Morphology and surface syntax: technical conversion
    - Penn Treebank style -> PDT Analytic layer
  - Tectogrammatical annotation: manual annotation
    - (Slightly) different rules needed for English
- Alignment
  - Natural, sentence level only (now)

Human Translation: WSJ Texts

- Hired translators / FCE level
- Specific rules for translation
  - Sentence per sentence only
    - …to get simple 1:1 alignment
  - Fluent Czech at the target side
  - If a choice, prefer “literal” translation
- The numbers:
  - English tokens: 1173766
  - Translated to Czech:
    - Revised/PCEDT 1.0: 487929
    - (now: 1097471)

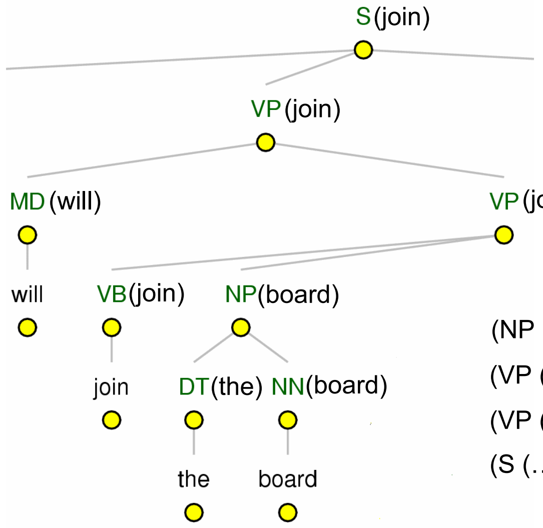
English Annotation
POS and Syntax

- Automatic conversion from Penn Treebank
  - PDT morphological layer
    - From POS tags
  - PDT analytic layer
    - From:
      - Penn Treebank Syntactic Structure
      - Non-terminal labels
      - Function tags (non-terminal “suffixes”)
    - 2-step process
      - Head determination rules
      - Conversion to dependency + analytic function

Head Determination Rules

- Exhaustive set of rules
  - By J. Eisner + M. Cmejrek/J. Curin
  - 4000 rules (non-terminal based)
    - Ex.: (S (NP-SBJ VP .)) → VP
  - Additional rules
    - Coordination, Apposition
    - Punctuation (end-of-sentence, internal)
- Original idea (possibility of conversion)
  - J. Robinson (1960s)

Example:
Head Determination Rules



Rules:

(NP (DT NN)) → NN

(VP (VB NP)) → VB

(VP (MD VP)) → VP

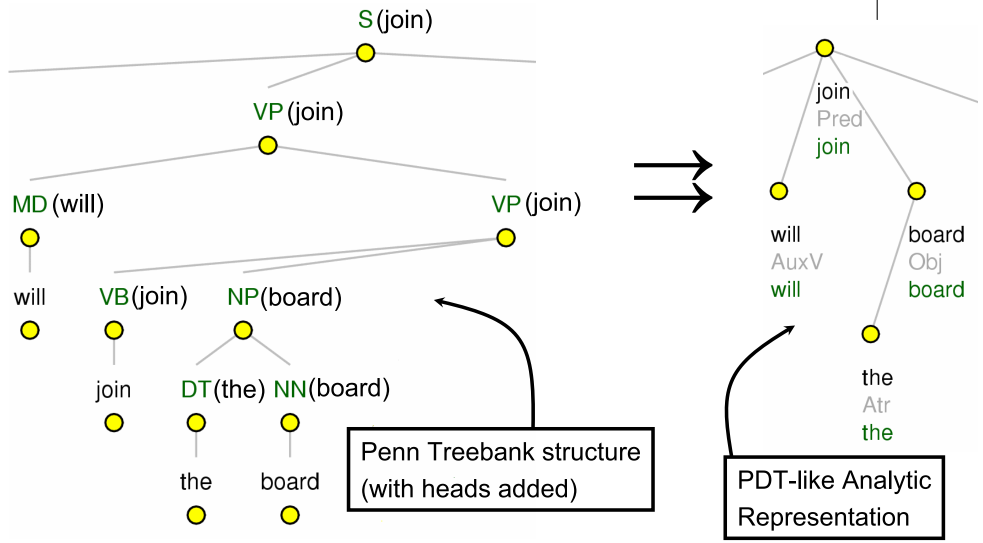
(S (… VP …)) → VP

Conversion to Analytic Structure, Functions

- Analytic Function assignment (conversion)
- Rules
  - based on functional tags:

    -SBJ Sb
    -BNF Obj
    -LGS Obj
    -DIR Adv
    -LOC Adv
    -PRP Adv
    -TMP Adv
    -PRD Pnom
    -DTV Obj
    -ADV Adv
    -EXT Adv
    -MNR Adv
    -PUT Adv
  - Ad-hoc rules (if functional tags missing)
  - Lemmatization (years → year)

Example: Conversion to Analytical Structure, Functions



Penn Treebank structure (with heads added)

PDT-like Analytic Representation

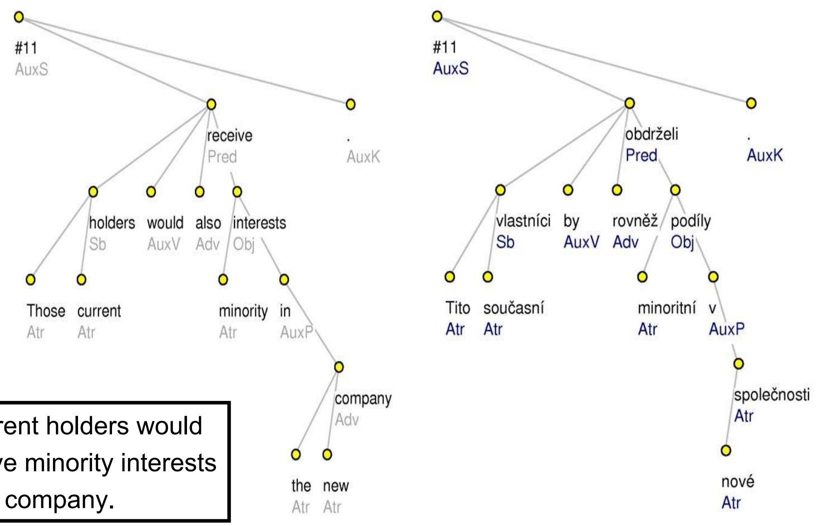
English PDT-style Annotation

- Morphology and Syntax
  - By conversion
- Tectogrammatical annotation
  - Manual
  - Pre-annotation
    - Transformation from Penn Treebank & Propbank (Palmer, Kingsbury)
  - Valency
    - From Propbank Frame Files
- Starting now

Czech PDT-style Annotation

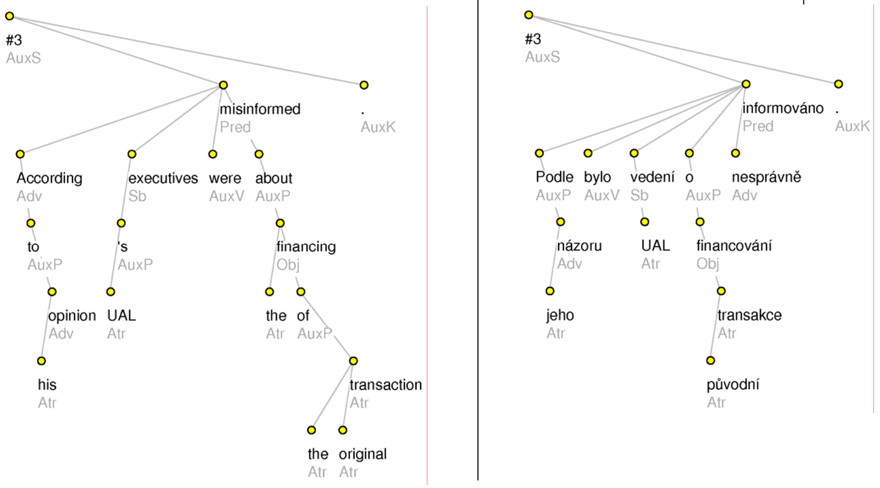
- All layers
  - (morphology, analytic, tectogrammatical)
- So far…
  - Automatic
- Manual annotation
  - Starting now
  - Top-down
    - Tectogrammatical first (lower layers automatically)
    - … then analytic structure and morphology

Analytical Pair En – Cz



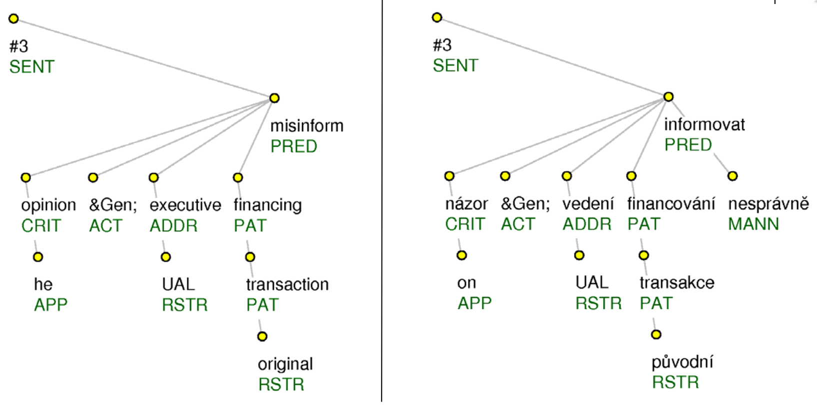
Those current holders would also receive minority interests in the new company.

Analytical Pair En - Cz



According to his opinion UAL's executives were misinformed about the financing of the original transaction.

Tectogrammatical Pair En - Cz



According to his opinion UAL's executives were misinformed about the financing of the original transaction.

Podle jeho názoru bylo vedení UAL o financování původní transakce nesprávně informováno.

Using Parallel Treebanks

- Word-based alignment
  - Phrasal elignment
- Dictionary extraction
  - From word/phrasal alignment
  - Probabilistic
- Machine translation
  - Statistical models
  - Evaluation/testing of systems

PCEDT 1.0 – The CD

- Published 2004 by the LDC (LDC2004T25)
- Texts, size of data:
  - 480,000 words: parallel annotated WSJ treebank
    - 21,600 sentences
  - 2 mil. words (53,000 sent.): Reader's Digest short stories
- Tools
  - GIZA++ (Statistical Machine Translation Toolkit)
  - Scripts for easy training ("SMT Quick Run")
  - Probabilistic dictionary (46,150 words, lemmatized)
    - Czech – English (WSJ and other sources)
- And more…

PCEDT – some pointers

- PCEDT 1.0
  - http://www.ldc.upenn.edu catalog No. LDC2004T25
  - http://ufal.mff.cuni.cz/pcedt
- PDT 2.0 (Czech annotation - documentation)
  - http://www.ldc.upenn.edu catalog No. LDC2006T01
  - http://ufal.mff.cuni.cz/pdt2.0
- Semecky, Cinkova:
  - Constructing an English Valency Lexicon
  - http://acl.ldc.upenn.edu/W/W06/W06-0612.pdf

Prague Treebanking for Everyone

Prague Arabic Dependency Treebank

Otakar Smrž

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

Vilem Mathesius Lecture Series 21

Prague, November 29, 2006

Prague Arabic Dependency Treebank

PADT is a project of linguistic annotation of Modern Written Arabic based on the theory of Functional Generative Description.

PADT 1.0 was published in 2004 by the Linguistic Data Consortium, and has been used by tens of academic and commercial institutions.

In PADT, which now consists of the morphological and the analytical levels of description of Arabic, the annotation of tectogrammatics and information structure is being established.

Outline

1. Introduction
2. Morphology
    - MorphoTrees
    - ElixirFM
3. Syntax and Beyond
    - Dependency Grammar
    - Analytical Syntax
    - Tectogrammatics
4. Software
    - TrEd Live
    - Encode Arabic
5. References

Outline

1. Introduction
2. Morphology
    - MorphoTrees
    - ElixirFM
3. Syntax and Beyond
    - Dependency Grammar
    - Analytical Syntax
    - Tectogrammatics
4. Software
    - TrEd Live
    - Encode Arabic
5. References

Morphology Disambiguation

Arabic is a language of rich morphology, both derivational and inflectional, with highly ambiguous orthography.

Boundaries of syntactic units, tokens, are obscure in writing—orthographical words, strings, consist of up to four lexemes.

Disambiguation encompasses subproblems like tokenization, full morphological tagging or its simplified 'part-of-speech' versions, lemmatization, diacritization or restoration of the structural components of words, plus combinations thereof.

He will notify them about that through SMS messages, the Internet, and other means. سَيُخبِرُهُم بِذٰلِكَ عَن طَرِيقِ الرَّسَائِلِ القَصِيرَةِ وَالإِنترنِتِ وَغَيرِهَا.

String · · Token	Token Tag	Buckwalter's M-Tags	Token Form	Token Gloss
سيخبرهم	`F---------`	FUT	*sa-*	will
	`VIIA-3MS--`	IV3MS+IV+IVSUFF_MOOD:I	*yu-ḫbir-u*	he-notify
	`S----3MP4-`	IVSUFF_DO:3MP	*-hum*	them
بذلك	`P---------`	PREP	*bi-*	about/by
	`SD----MS--`	DEM_PRON_MS	*ḏālika*	that
عن	`P---------`	PREP	*ʿan*	by/about
طريق	`N-------2R`	NOUN+CASE_DEF_GEN	*ṭarīq-i*	way-of
الرسائل	`N-------2D`	DET+NOUN+CASE_DEF_GEN	*ar-rasāʾil-i*	the-messages
القصيرة	`A-----FS2D`	DET+ADJ+NSUFF_FEM_SG+ +CASE_DEF_GEN	*al-qaṣīr-at-i*	the-short
والإنترنت	`C---------`	CONJ	*wa-*	and
	`Z-------2D`	DET+NOUN_PROP+ +CASE_DEF_GEN	*al-ʾinternet-i*	the-internet
وغيرها	`C---------`	CONJ	*wa-*	and
	`FN------2R`	NEG_PART+CASE_DEF_GEN	*ġayr-i*	other/not-of
	`S----3FS2-`	POSS_PRON_3FS	*-hā*	them

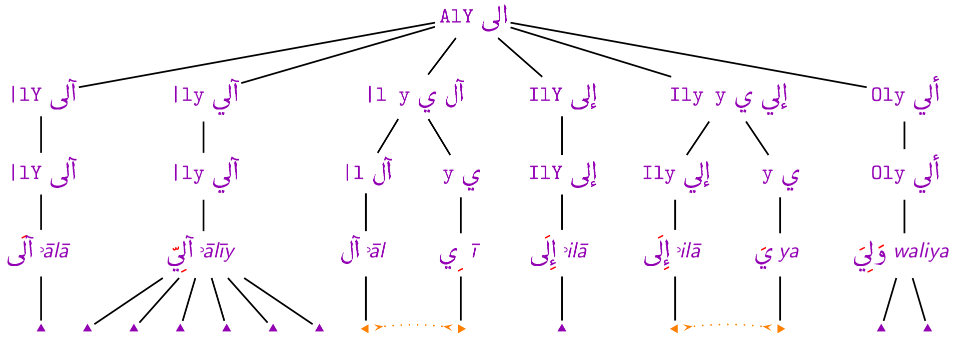
MorphoTrees

Suppose you can list morphological analyses for a given input string …

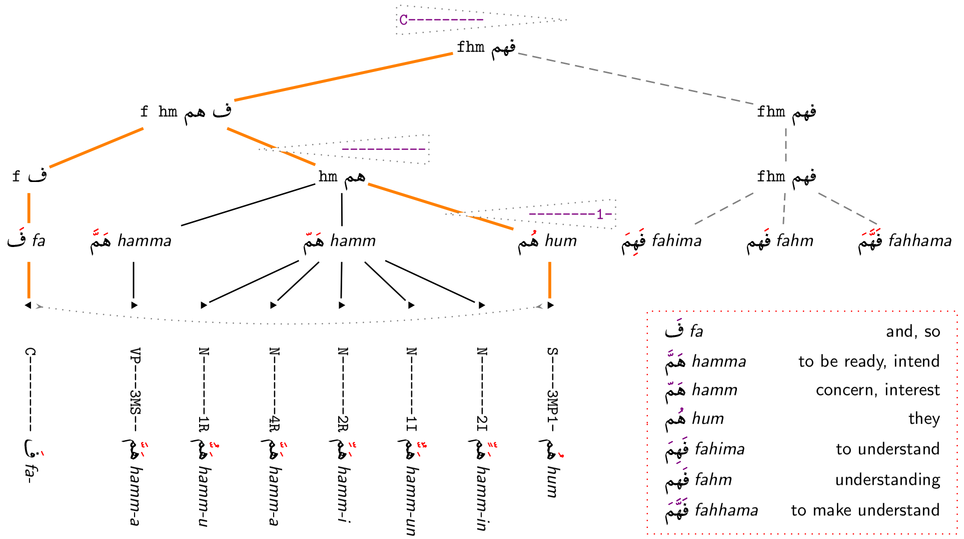
Morphs	Form	Token Tag	Lemma	Glosses per Morph
|laY+(null)	ʾālā	VP-A-3MS--	ʾālā	promise/take an oath + he/it
|liy~	ʾālīy	A---------	ʾālīy	mechanical/automatic
|liy~+u	ʾālīy-u	A-------1R	ʾālīy	mechanical … + [def.nom.]
|liy~+i	ʾālīy-i	A-------2R	ʾālīy	mechanical … + [def.gen.]
|liy~+a	ʾālīy-a	A-------4R	ʾālīy	mechanical … + [def.acc.]
|liy~+N	ʾālīy-un	A-------1I	ʾālīy	mechanical … + [indef.nom.]
|liy~+K	ʾālīy-in	A-------2I	ʾālīy	mechanical … + [indef.gen.]
|l +	ʾāl	N--------R	ʾāl	family/clan +
+ iy	-ī	S----1-S2-	ī	+ my
IilaY	ʾilā	P---------	ʾilā	to/towards
Iilay +	ʾilay	P---------	ʾilā	to/towards +
+ ya	-ya	S----1-S2-	ya	+ me
Oa+liy+(null)	ʾa-lī	VIIA-1-S--	waliya	I + follow/come after + [ind.]
Oa+liy+a	ʾa-liy-a	VISA-1-S--	waliya	I + follow/come after + [sub.]

MorphoTrees

… organize the analyses into a hierarchy with the string as its root and the full tokens as the leaves, grouped by their lemmas, canonical forms and partitionings of the string into such forms:



MorphoTrees



فَ fa	and, so
هَمَّ hamma	to be ready, intend
هَمّ hamm	concern, interest
هُم hum	they
فَهِمَ fahima	to understand
فَهم fahm	understanding
فَهَّمَ fahhama	to make understand

ElixirFM

ElixirFM is a high-level implementation of Functional Arabic Morphology.

ElixirFM uses the Functional Morphology library for Haskell and extends it.

Morphology is modeled in terms of paradigms, grammatical categories, lexemes and word classes. The computation of analysis or generation is conceptually distinguished from the general-purpose linguistic model.

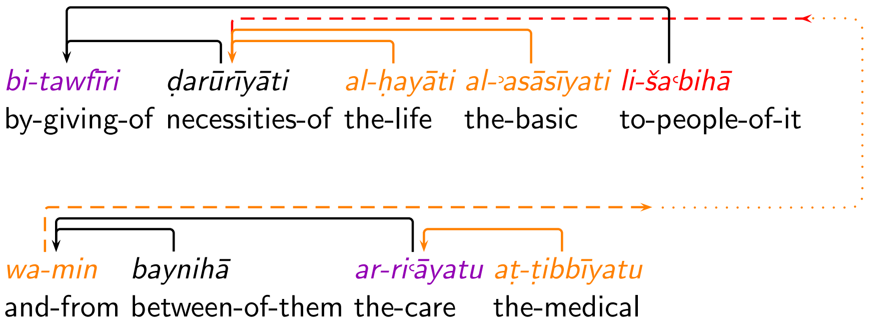
The lexicon of ElixirFM is derived from the open-source Buckwalter lexicon and from the PADT annotations. It is redesigned in important respects.

Outline

1. Introduction
2. Morphology
    - MorphoTrees
    - ElixirFM
3. Syntax and Beyond
    - Dependency Grammar
    - Analytical Syntax
    - Tectogrammatics
4. Software
    - TrEd Live
    - Encode Arabic
5. References

Dependency vs. Linearity

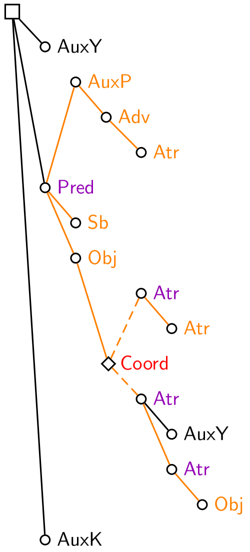
. . . by providing the basic necessities of life to its people, including medical care . . . بِتَوفِيرِ ضَرُورِيَّاتِ الحَيَاةِ الأَسَاسِيَّةِ لِشَعبِهَا وَمِن بَينِهَا الرِّعَايَةُ الطِّبِّيَّةُ



bi-tawfīri *ḍarūrīyāti* *al-ḥayāti* *al-ʾasāsīyati* *li-šaʿbihā*
by-giving-of necessities-of the-life the-basic to-people-of-it

wa-min *baynihā* *ar-riʿāyatu* *aṭ-ṭibbīyatu*
and-from between-of-them the-care the-medical

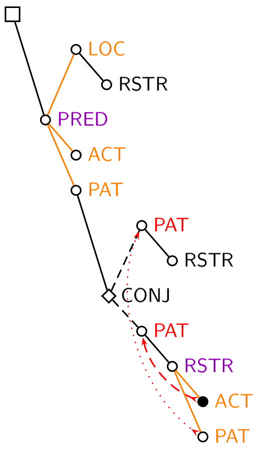
In the section on literature, the magazine presented the issue of the Arabic language and the dangers that threaten it. وَفِي مِلَفِّ الأَدَبِ طَرَحَتِ . . .



وَ *wa-*	and	`C---------`
فِي *fī*	in	`P---------`
مِلَفِّ *milaffi*	collection/file-of	`N-------2R`
اَلأَدَبِ *al-ʾadabi*	the-literature	`N-------2D`
طَرَحَت *ṭaraḥat*	it-presented	`VP-A-3FS--`
اَلمَجَلَّةُ *al-maǧallatu*	the-magazine	`N-----FS1D`
قَضِيَّةَ *qaḍīyata*	issue-of	`N-----FS4R`
اَللُّغَةِ *al-luġati*	the-language	`N-----FS2D`
اَلعَرَبِيَّةِ *al-ʿarabīyati*	the-Arabic	`A-----FS2D`
وَ *wa-*	and	`C---------`
اَلأَخطَارِ *al-ʾaḫṭāri*	the-dangers	`N-------2D`
اَلَّتِي *allatī*	that	`SR----FS--`
تُهَدِّدُ *tuhaddidu*	they-threaten	`VIIA-3FS--`
هَا *-hā*	it	`S----3FS4-`
. .	.	`G---------`

Tectogrammatics

Description of linguistic meaning in its semantic and pragmatic aspects.



B	ملف *milaff*	collection/file	Masc.Sing.Def
C	أدب *ʾadab*	literature	Masc.Sing.Def
B	طرح *ṭaraḥ*	to-present	Ind.Ant.Act
B	مجلة *maǧallah*	magazine	Fem.Sing.Def
N	قضية *qaḍīyah*	issue	Fem.Sing.Def
N	لغة *luġah*	language	Fem.Sing.Def
N	عربي *ʿarabīy*	Arabic	Adjective
	و *wa*	and	Coordination
N	خطر *ḫaṭar*	danger	Masc.Plur.Def
N	هدد *haddad*	to-threaten	Ind.Sim.Act
B	هي *hiya*	it	PersPronoun
B	هي *hiya*	it	PersPronoun

Outline

1. Introduction
2. Morphology
    - MorphoTrees
    - ElixirFM
3. Syntax and Beyond
    - Dependency Grammar
    - Analytical Syntax
    - Tectogrammatics
4. Software
    - TrEd Live
    - Encode Arabic
5. References

Buckwalter Transliteration

يُولَدُ جَمِيعُ ٱلنَّاسِ أَحْرَارًا مُتَسَاوِينَ فِي ٱلْكَرَامَةِ وَٱلْحُقُوقِ. وَقَدْ وُهِبُوا عَقْلًا وَضَمِيرًا وَعَلَيْهِمْ أَنْ يُعَامِلَ بَعْضُهُمْ بَعْضًا بِرُوحِ ٱلْإِخَاءِ.

```
yuwladu jamiyEu  {ln~aAsi OaHoraArFA mutasaAwiyna fiy
{lokaraAmapi wa {loHuquwqi.  waqado wuhibuwA EaqolAF
waDamiyrFA waEalayohimo Oano yuEaAmila baEoDuhumo baEoDFA
biruwHi  {loIixaA'i.
```

يولد جميع الناس أحرارا متساوين في الكرامة والحقوق. وقد وهبوا عقلا وضميرا وعليهم أن يعامل بعضهم بعضا بروح الإخاء.

```
ywld jmyE AlnAs OHrArA mtsAwyn fy AlkrAmp wAlHqwq.  wqd
whbwA EqlA wDmyrA wElyhm On yEAml bEDhm bEDA brwH AlIxA'.
```

Notation of ArabTEX

يُولَدُ جَمِيعُ ٱلنَّاسِ أَحْرَارًا مُتَسَاوِينَ فِي ٱلْكَرَامَةِ وَٱلْحُقُوقِ. وَقَدْ وُهِبُوا عَقْلًا وَضَمِيرًا وَعَلَيْهِمْ أَنْ يُعَامِلَ بَعْضُهُمْ بَعْضًا بِرُوحِ ٱلْإِخَاءِ.

يولد جميع النّاس أحرارا متساوين في الكرامة والحقوق. وقد وهبوا عقلا وضميرا وعليهم أن يعامل بعضهم بعضا بروح الإخاء.

Yūladu ǧamīʿu ʾn-nāsi ʾaḥrāran mutasāwīna fī ʾl-karāmati wa-ʾl-ḥuqūqi. Wa-qad wuhibū ʿaqlan wa-ḍamīran wa-ʿalayhim ʾan yuʿāmila baʿḍuhum baʿḍan bi-rūḥi ʾl-ʾiḫāʾi.

```
\cap yUladu ^gamI‘u an-nAsi ’a.hrAraN mutasAwIna fI
     al-karAmaTi wa-al-.huqUqi.
\cap wa-qad wuhibUA ‘aqlaN wa-.damIraN wa-‘alayhim ’an
     yu‘Amila ba‘.duhum ba‘.daN bi-rU.hi al-’i_hA’i.
```

Encode Arabic

biruwHi {loIixaA'i ← بِرُوحِ ٱلْإِخَاءِ ← bi-rU.hi al-'i_hA'i

Implemented in Perl and available on CPAN as Encode-Arabic:

```
$encoded = encode "buckwalter", decode "arabtex", $decoded
$encoded = encode("buckwalter", decode("arabtex", $decoded))
```

Implemented in Haskell and available along with ElixirFM:

```
encoded = encode Buckwalter $ decode ArabTeX decoded
encoded = encode Buckwalter (decode ArabTeX decoded)
encoded = (encode Buckwalter . decode ArabTeX) decoded
```

```
[cmd] decode ArabTeX < decode.d | encode Buckwalter > encode.d
```

Outline

1. Introduction
2. Morphology
    - MorphoTrees
    - ElixirFM
3. Syntax and Beyond
    - Dependency Grammar
    - Analytical Syntax
    - Tectogrammatics
4. Software
    - TrEd Live
    - Encode Arabic
5. References

- Buckwalter, Tim. Buckwalter Arabic Morphological Analyzer 1.0. LDC catalog number LDC2002L49, ISBN 1-58563-257-0. 2002
- Forsberg, Markus and Aarne Ranta. Functional Morphology. Proceedings of ICFP 2004, pages 213–223. ACM Press. 2004
- Lagally, Klaus. ArabTeX: Typesetting Arabic and Hebrew, User Manual Version 4.00. Technical Report 2004/03, Fakultät Informatik, Universität Stuttgart. 2004
- Sgall, Petr and Eva Hajičová and Jarmila Panevová. The Meaning of the Sentence in Its Semantic and Pragmatic Aspects. Academia, Prague. 1986
- Smrž, Otakar and Petr Pajas. MorphoTrees of Arabic and Their Annotation in the TrEd Environment. Proceedings of the NEMLAR Conference 2004, pages 38–41. 2004

PADT++ `http://ufal.mff.cuni.cz/padt/online/`

CZECH POSITIONAL MORPHOLOGICAL TAGS

1. PART OF SPEECH

A	Adjectives
C	Numerals
D	Adverbs
I	Interjection
J	Conjunction
N	Noun
P	Pronoun
V	Verb
R	Preposition
T	Particle
X	Unknown, Not Determined, Unclassifiable
Z	Punctuation (also used for the Sentence Boundary Token)

2. SUB PART OF SPEECH

#	Sentence boundary
%	Author's signature, e.g. haš-99_:B_;S
*****	Word krát (lit.: "times")
,	Conjunction subordinate (incl. "aby", "kdyby" in all forms)
}	Numeral, written using Roman numerals (XIV)
:	Punctuation (except for the virtual sentence boundary word ###, which uses "Sub part of speech" #)
=	Number written using digits
?	Numeral "kolik" (lit. "how many"/"how much")
@	Unrecognized word form
^	Conjunction (connecting main clauses, not subordinate)
4	Relative/interrogative pronoun with adjectival declension of both types (soft and hard) ("jaký", "který", "čí", ..., lit. "what", "which", "whose", ...)
5	The pronoun he in forms requested after any preposition (with prefix n-: "něj", "něho", ..., lit. "him" in various cases)
6	Reflexive pronoun "se" in long forms ("sebe", "sobě", "sebou", lit. "myself" / "yourself" / "herself" / "himself" in various cases; "se" is personless)
7	Reflexive pronouns "se" ("Case" = 4), "si" ("Case" = 3), plus the same two forms with contracted -s: "ses", "sis" (distinguished by "Person" = 2; also number is singular only) This should be done somehow more consistently, virtually any word can have this contracted -s ("cos", "polívkus", ...)
8	Possessive reflexive pronoun "svůj" (lit. "my"/"your"/"her"/"his" when the possessor is the subject of the sentence)
9	Relative pronoun "jenž", "již", ... after a preposition (n-: "něhož", "niž", ..., lit. "who")
A	Adjective, general
B	Verb, present or future form
C	Adjective, nominal (short, participial) form "rád", "schopen", ...
D	Pronoun, demonstrative ("ten", "onen", ..., lit. "this", "that", "that", ... "over there", ...)
E	Relative pronoun "což" (corresponding to English which in subordinate clauses referring to a part of the preceding text)
F	Preposition, part of; never appears isolated, always in a phrase ("nehledě (na)", "vzhledem (k)", ..., lit. "regardless", "because of")
G	Adjective derived from present transgressive form of a verb
H	Personal pronoun, clitical (short) form ("mě", "mi", "ti", "mu", ...); these forms are used in the second position in a clause (lit. "me", "you", "her", "him"), even though some of them ("mě") might be regularly used anywhere as well
I	Interjections
J	Relative pronoun "jenž", "již", ... not after a preposition (lit. "who", "whom")
K	Relative/interrogative pronoun "kdo" (lit. "who"), incl. forms with affixes -ž and -s (affixes are distinguished by the category "Variant" (for -ž) and "Person" (for -s))
L	Pronoun, indefinite "všechen", "sám" (lit. "all", "alone")
M	Adjective derived from verbal past transgressive form
N	Noun (general)
O	Pronoun "svůj", "nesvůj", "tentam" alone (lit. "own self", "not-in-mood", "gone")
P	Personal pronoun "já", "ty", "on" (lit. "I", "you", "he") (incl. forms with the enclitic -s, e.g. "tys", lit. "you're"); gender position is used for third person to distinguish "on"/"ona"/"ono" (lit. "he"/"she"/"it"), and number for all three persons
Q	Pronoun relative/interrogative "co", "copak", "cožpak" (lit. "what", "isn't-it-true-that")
R	Preposition (general, without vocalization)
S	Pronoun possessive "můj", "tvůj", "jeho" (lit. "my", "your", "his"); gender position used for third person to distinguish "jeho", "její", "jeho" (lit. "his", "her", "its"), and number for all three pronouns
T	Particle
U	Adjective possessive (with the masculine ending -ův as well as feminine -in)
V	Preposition (with vocalization -e or -u): ("ve", "pode", "ku", ..., lit. "in", "under", "to")
W	Pronoun negative ("nic", "nikdo", "nijaký", "žádný", ..., lit. "nothing", "nobody", "not-worth-mentioning", "no"/"none")
X	(temporary) Word form recognized, but tag is missing in dictionary due to delays in (asynchronous) dictionary creation
Y	Pronoun relative/interrogative co as an enclitic (after a preposition) ("oč", "nač", "zač", lit. "about what", "on"/"onto" "what", "after"/"for what")
Z	Pronoun indefinite ("nějaký", "některý", "čízkoli", "cosi", ..., lit. "some", "some", "anybody's", "something")
a	Numeral, indefinite ("mnoho", "málo", "tolik", "několik", "kdovíkolik", ..., lit. "much"/"many", "little"/"few", "that much"/"many", "some" ("number of"), "who-knows-how-much/many")
b	Adverb (without a possibility to form negation and degrees of comparison, e.g. "pozadu", "naplocho", ..., lit. "behind", "flatly"); i.e. both the "Negation" as well as the "Grade" attributes in the same tag are marked by – (Not applicable)
c	Conditional (of the verb "být" (lit. "to be") only) ("by", "bych", "bys", "bychom", "byste", lit. "would")
d	Numeral, generic with adjectival declension ("dvojí", "desaterý", ..., lit. "two-kinds"/..., "ten-...")
e	Verb, transgressive present (endings -e/-ě, -íc, -íce)
f	Verb, infinitive
g	Adverb (forming negation ("Negation" set to A/N) and degrees of comparison "Grade" set to 1/2/3 (comparative/superlative), e.g. "velký", "za\-jí\-ma\-vý", ..., lit. "big", "interesting"
h	Numeral, generic: only "jedny" and "nejedny" (lit. "one-kind"/"sort-of", "not-only-one-kind"/"sort-of")
i	Verb, imperative form
j	Numeral, generic greater than or equal to 4 used as a syntactic noun ("čtvero", "desatero", ..., lit. "four-kinds"/"sorts-of", "ten-...")
k	Numeral, generic greater than or equal to 4 used as a syntactic adjective, short form ("čtvery", ..., lit. "four-kinds"/"sorts-of")
l	Numeral, cardinal "jeden", "dva", "tři", "čtyři", "půl", ... (lit. "one", "two", "three", "four"); also "sto" and "tisíc" (lit. "hundred", "thousand") if noun declension is not used
m	Verb, past transgressive; also archaic present transgressive of perfective verbs (ex.: "udělav", lit. "(he-)having-done"; arch. also "udělaje" ("Variant" = 4), lit. "(he-)having-done)"
n	Numeral, cardinal greater than or equal to 5
o	Numeral, multiplicative indefinite ("-krát", lit. ("times"): "mnohokrát", "tolikrát", ..., lit. "many times", "that many times")
p	Verb, past participle, active (including forms with the enclitic - s, lit. 're ("are"))
q	Verb, past participle, active, with the enclitic -ť, lit. ("perhaps") - "could-you-imagine-that?" or "but-because-" (both archaic)
r	Numeral, ordinal (adjective declension without degrees of comparison)
s	Verb, past participle, passive (including forms with the enclitic -s, lit. 're ("are"))
t	Verb, present or future tense, with the enclitic -ť, lit. ("perhaps") "-could-you-imagine-that?" or "but-because-" (both archaic)
u	Numeral, interrogative "kolikrát", lit. "how many times?"
v	Numeral, multiplicative, definite (-krát, lit. "times": "pětkrát", ..., lit. "five times")
w	Numeral, indefinite, adjectival declension ("nejeden", "tolikátý", ..., lit. "not-only-one", "so-many-times-repeated")
y	Numeral, fraction ending at -ina; used as a noun ("pětina", lit. "one-fifth")
z	Numeral, interrogative "kolikátý", lit. "what" ("at-what-position-place-in-a-sequence")

3. GENDER

F	Feminine
H	{F, N} – Feminine or Neuter
I	Masculine inanimate
M	Masculine animate
N	Neuter
Q	Feminine (with singular only) or Neuter (with plural only); used only with participles and nominal forms of adjectives
T	Masculine inanimate or Feminine (plural only); used only with participles and nominal forms of adjectives
X	Any
Y	{M, I} – Masculine (either animate or inanimate)
Z	{M, I, N} – Not feminine (i.e., Masculine animate/inanimate or Neuter); only for (some) pronoun forms and certain numerals

4. NUMBER

D	Dual , e.g. "nohama"
P	Plural, e.g. "nohami"
S	Singular, e.g. "noha"
W	Singular for feminine gender, plural with neuter; can only appear in participle or nominal adjective form with gender value Q
X	Any

5. CASE

1	Nominative, e.g. "žena"
2	Genitive, e.g. "ženy"
3	Dative, e.g. "ženě"
4	Accusative, e.g. "ženu"
5	Vocative, e.g. "ženo"
6	Locative, e.g. "ženě"
7	Instrumental, e.g. "ženou"
X	Any

6. POSSESSIVE GENDER

F	Feminine, e.g. "matčin", "její"
M	Masculine animate (adjectives only), e.g. "otců"
X	Any
Z	{M, I, N} – Not feminine, e.g. "jeho"

7. POSSESSIVE NUMBER

P	Plural, e.g. "náš"
S	Singular, e.g. "můj"
X	Any, e.g. "your"

8. PERSON

1	1st person, e.g. "píšu", "píšeme"
2	2nd person, e.g. "píšeš", "píšete"
3	3rd person, e.g. "píše", "píšou"
X	Any person

9. TENSE

F	Future
H	{R, P} – Past or Present
P	Present
R	Past
X	Any

10. GRADE

1	Positive, e.g. "velký"
2	Comparative, e.g. "větší"
3	Superlative, e.g. "největší"

11. NEGATION

A	Affirmative (not negated), e.g. "možný"
N	Negated, e.g. "nemožný"

12. VOICE

A	Active, e.g. "píšící"
P	Passive, e.g. "psaný"

13., 14. RESERVE 1, RESERVE 2

-	Not applicable

15. VARIANT

-	Basic variant, standard contemporary style; also used for standard forms allowed for use in writing by the Czech Standard Orthography Rules despite being marked there as colloquial
1	Variant, second most used (less frequent), still standard
2	Variant, rarely used, bookish, or archaic
3	Very archaic, also archaic + colloquial
4	Very archaic or bookish, but standard at the time
5	Colloquial, but (almost) tolerated even in public
6	Colloquial (standard in spoken Czech)
7	Colloquial (standard in spoken Czech), less frequent variant
8	Abbreviations
9	Special uses, e.g. personal pronouns after prepositions etc.

Analytical functions in PDT 2.0

Pred	predicate, a node not depending on another node; depends on #	**Pnom**	nominal predicate, or nom. part of predicate with copula *be*	**AuxC**	conjunction (subord.)	**AuxK**	terminal punctuation of a sentence
Sb	subject	**AuxV**	auxiliary verb *be*	**AuxO**	redundant or emotional item, ‘coreferential’ pronoun	**ExD**	a technical value for a deleted item; also for the main element of a sentence without predicate (externally-dependent)
Obj	object	**Coord**	coord. node	**AuxZ**	emphasizing word	**AtrAtr**	an attribute of any several preceding (syntactic) nouns
Adv	adverbial	**Apos**	apposition (main node)	**AuxX**	comma (not serving as a coordinating conjunction)	**AtrAdv**	structural ambiguity between adverbial and adnominal (hung on a name/noun) dependency without a semantic difference
Atv	complement (so-called determining) technically hung on a non-verbal element	**AuxT**	reflexive tantum	**AuxG**	other graphic symbols, not terminal	**AdvAtr**	dtto with reverse preference
AtvV	complement (so-called determining) hung on a verb, no 2nd gov. node	**AuxR**	passive reflexive	**AuxY**	adverbs, particles not classed elsewhere	**AtrObj**	structural ambiguity between object and adnominal dependency without a semantic difference
Atr	attribute	**AuxP**	primary preposition, parts of a secondary preposition	**AuxS**	root of the tree (#)	**ObjAtr**	dtto with reverse preference

T-node attributes and their values in PDT 2.0

Notation:
attribute – attribute name
value – attribute value

A. Lexical content

t_lemma – tectogrammatical lemma

val_frame.rf – valency frame (reference to PDT-VALLEX)

B. Semantic roles and other structural relations

functor – role of the node within the t-tree structure

Functors for independent clauses:
1. PRED – predicate clause
2. DENOM – denominative clause
3. VOCAT – vocative clause
4. PARTL – interjectional clause
5. PAR – parenthetical clause

Actants:
6. ACT – actor
7. PAT – patient
8. ADDR – addressee
9. ORIG – origin
10. EFF – effect

Temporal modifiers:
11. TWHEN – when
12. TFHL – for how long
13. TFRWH – from when
14. THL – how long
15. THO – how often
16. TOWH – to when
17. TPAR – temporal parallel
18. TSIN – since when
19. TTILL – till when

Spatial modifiers:
20. LOC – where
21. DIR1 – from where
22. DIR2 – through where
23. DIR3 – to where

Implicational/causal modifiers:
24. AIM – aim
25. CAUS – cause
26. CNCS – concession
27. COND – condition
28. INTT – intention

Various types of manner:
29. ACMP – accompaniment
30. CPR – comparison
31. CRIT – criterion
32. DIFF – difference
33. EXT – extent
34. MANN – manner
35. MEANS – means
36. REG – regard
37. RESL – result
38. RESTR – restriction

Specific adnominal modifiers:
39. RSTR – attribute
40. APP – appurtenance
41. AUTH – author
42. MAT – material
43. ID – identity

Paratactic structures:
44. ADVS – adversative
45. CONFR – confrontation
46. CONJ – conjunction
47. CONTRA – contrariety
48. CSQ – consequence
49. DISJ – disjunction
50. GRAD – gradation
51. REAS – reason
52. APPS – apposition
53. CM – coordination modifier

Multiword lexical units:
54. CPHR – part of complex predicate
55. DPHR – dependent part of an idiomatic expression

Other:
56. COMPL – predicative complement
57. BEN – benefactor
58. CONTRD – contradiction
59. HER – heritage
60. RHEM – rhematizer
61. SUBS – substitution
62. ATT – attitude
63. INTF – intensifier
64. MOD – modality
65. PREC – reference to preceding text
66. FPHR – foreign language expression

subfunctor – more detailed functor specification
1. basic – basic value (prototypical for the given functor)
2. nr – not recognized

Values specific to spatial functors:
3. abstr – in abstract space
4. along – along
5. around – around
6. above – above
7. behind – behind
8. below – below
9. betw – between
10. elsew – elsewhere
11. ext – extent
12. front – in front of
13. near – near
14. opp – opposite
15. target – target
16. to – to
17. across – across

Values specific to ACMP:
18. circ – circumstance
19. incl – inclusion
20. wout – negative accompaniment (without someone)

Values specific to CPR:
21. than – difference
22. wrt – with respect to

Values specific to BEN:
23. agst – against

Values specific to EXT:
24. approx – approximately
25. less – less
26. more – more

Values specific to TWHEN:
27. after – after
28. approx – approximately
29. before – before
30. begin – at the beginning of
31. betw – between
32. end – at the end of
33. flow – in the course of
34. mid – in the middle of

is_member – distinction between members of paratactic structures and shared modifiers
1. 0 – non-member
2. 1 – member

is_parenthesis
1. 0 – unmarked value
2. 1 – part of parenthesis

is_state
1. 0 – unmarked value
2. 1 – modifier expressing being in certain state

C. Communicative dynamism

tfa – topic/focus articulation
1. t – non-contrastive contextually bound expression
2. f – contextually non-bound expression
3. c – contrastive contextually bound expression

deepord – non-negative integer representing deep word order

D. Coreference and predicative complement

coref_gram.rf – (list of) reference(s) to antecedent(s) in the sense of grammatical coreference

coref_text.rf – (list of) reference(s) to antecedent(s) in the sense of textual coreference

coref_special – special types of coreference (without obvious t-node antecedent)
1. segm – coreference with a sequence of preceding sentences, without more explicit limitations
2. exoph – antecedent not present in the text at all

compl.rf – reference to "secondary" parent t-node (in the case of "dual" complement dependency)

E. Types of t-nodes

nodetype – basic node classification
1. root – technical root
2. complex – complex node
3. qcomplex – quasi-complex node
4. atom – atomic node
5. coap – paratactic structure root (coordination or apposition)
6. dphr – dependent part of an idiomatic expression
7. fphr – part of a foreign-language expression
8. list – root node of a list structure

sempos – semantic part of speech (further subdivision of complex nodes)
1. n.denot – denotative semantic noun
2. n.denot.neg – denotative semantic noun with separately represented negation
3. n.pron.def.demon – demonstrative definite pronominal semantic noun
4. n.pron.def.pers – personal definite pronominal semantic noun
5. n.pron.indef – indefinite pronominal semantic noun
6. n.quant.def – definite quantificational semantic noun
7. adj.denot – denotative semantic adjective
8. adj.pron.def.demon – demonstrative definite pronominal semantic adjective
9. adj.pron.indef – indefinite pronominal semantic adjective
10. adj.quant.def – definite quantificational semantic adjective
11. adj.quant.indef – indefinite quantificational semantic adjective
12. adj.quant.grad – gradable quantificational semantic adjective
13. adv.denot.ngrad.nneg – non-gradable denotative semantic adverb, impossible to negate
14. adv.denot.ngrad.neg – non-gradable denotative semantic adverb, possible to negate
15. adv.denot.grad.nneg – gradable denotative semantic adverb, impossible to negate
16. adv.denot.grad.neg – gradable denotative semantic adverb, possible to negate
17. adv.pron.def – definite pronominal semantic adverb
18. adv.pron.indef – indefinite pronominal semantic adverb
19. v – semantic verb

F. Grammatemes

sentmod – sentence modality
1. enunc – indicative mood
2. excl – exclamation mood
3. desid – desiderative mood
4. imper – imperative mood
5. inter – interrogative mood

value applicable to all following grammatemes:
nr – not recognized

gram/aspect – aspect
1. proc – processual (counterpart to imperfective)
2. cpl – complex (counterpart to perfective)

gram/degcmp – degree of comparison
1. pos – positive
2. comp – comparative
3. acomp – absolute comparative
4. sup – superlative

gram/deontmod – deontic modality
1.deb – necessary
2. hrt – obligatory
3. vol – wanted/intended
4. poss – possible
5. perm – permitted
6. fac – ability to do something
7. decl – unmarked

gram/dispmod – dispositional modality
1. disp0 – dispositional modality absent
2. disp1 – dispositional modality present
3. nil – not applicable (with infinitive)

gram/gender – gender
1. anim – masculine animate
2. inan – masculine inanimate
3. fem – feminine
4. neut – neuter
5. inher – "inherited" from antecedent

gram/indeftype – type of (pro-form) indefiniteness
1. relat – relative
2. inter – interrogative
3. negat – negative
4.-10. indef1 – indef6 – other types of indefiniteness
11.-12. total1, total2 – totalizers

gram/iterativeness – iterativeness
1. it0 – non-iterative verb
2. it1 – iterative verb

gram/negation – negation
1. neg0 – affirmative
2. neg1 – negative

gram/number – number
1. sg – singular
2. pl – plural
3. inher – "inherited" from antecedent

gram/numertype – type of numeral expression
1. basic – basic numeral
2. frac – fractional numeral
3. kind – sort numeral
4. ord – ordinal numeral
5. set – set numeral

gram/person – person
1. 1 – first person
2. 2 – second person
3. 3 – third person
4. inher – "inherited" from antecedent

gram/politeness – politeness
1. basic – common use
2. polite – polite form

gram/resultative – resultative
1. res0 – non-resultative
2. res1 – resultative

gram/tense – verb tense
1. sim – simultaneous
2. ant – preceding (anterior)
3. post – subsequent (posterior)
4. nil – not applicable (with infinitive)

gram/verbmod – verb modality
1. ind – indicative
2. imp – imperative
3. cdn – conditional
4. nil – not applicable (with infinitive)

G. Links to a-layer

atree.rf – reference to the corresponding a-tree technical root (only with technical t-tree root)

a/lex.rf – reference to (identifier of) the corresponding "autosemantic" a-node

a/aux.rf – (list of) reference(s) to the corresponding auxiliary a-node(s)

is_generated – distinction between nodes expressed/unexpressed in the surface form
1. 0 – surface counterpart exists
2. 1 – newly created (or "copied") node

H. Quotation and direct speech

quot/type – type of quoted expression
1. citation – citation
2. dsp – direct speech
3. meta – "meta" use
4. title – title
5. other – other type

quot/set_id – id dedicated for co-indexing all nodes within a quoted expression

is_dsp_root – root of direct speech
1. 0 – unmarked value
2. 1 – root of subtree representing direct speech

I. Other

id – node identifier

is_name_of_person – personal proper name
1. 0 – unmarked value
2. 1 – proper name of a person