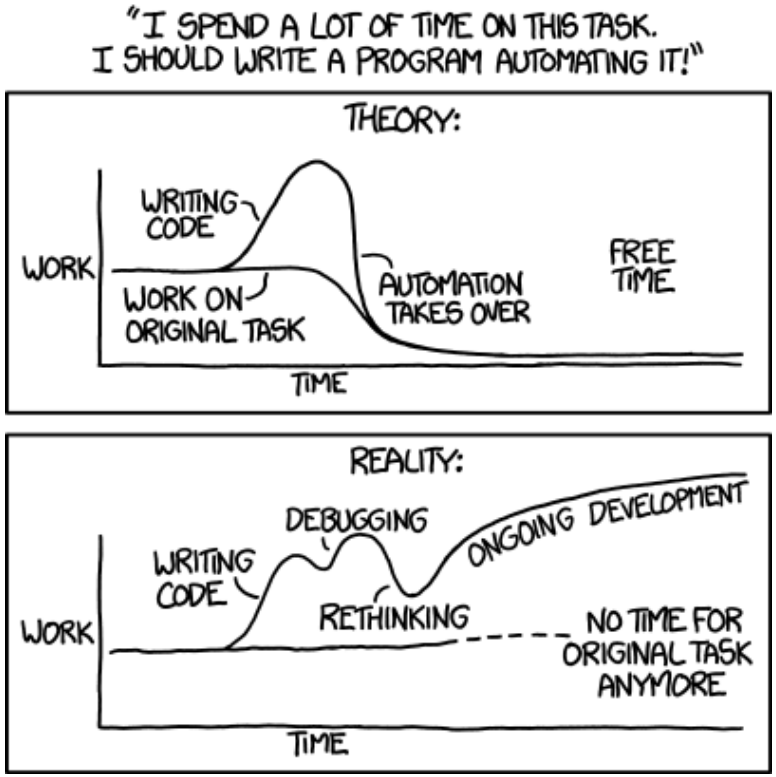
Working with Universal Dependencies

Martin Popel
popel@ufal.mff.cuni.cz

ÚFAL (Institute of Formal and Applied Linguistics)
Charles University, 2017-03-13



https://xkcd.com/1319/

Overview

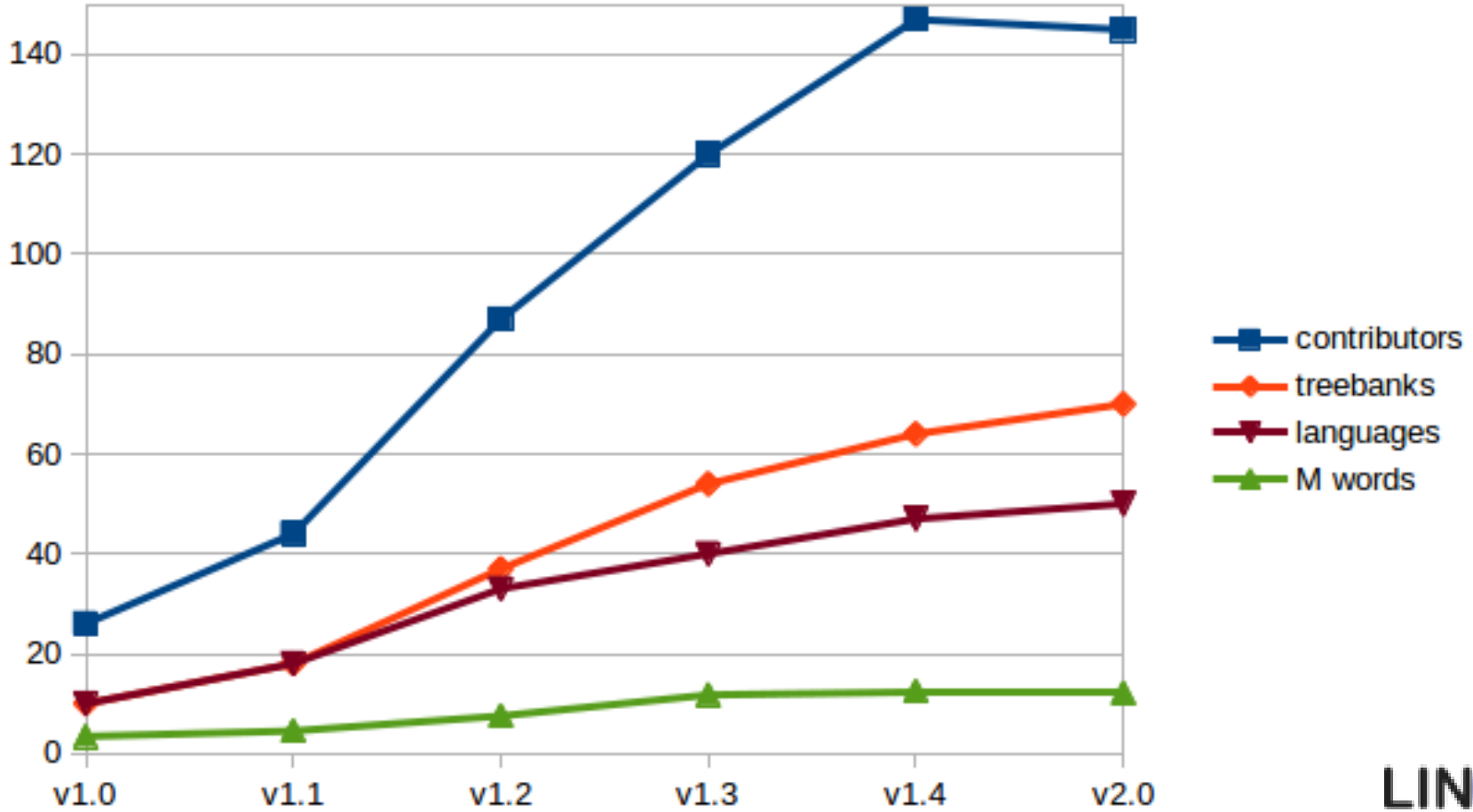
- Universal Dependencies
- A thought on simplicity
- CoNLL 2017 Shared Task
- Tools for UD
- Udapi

Universal Dependencies

- cross-linguistically consistent treebank annotation
- de facto standard for dependency annotation
- builds on:
  Stanford Dependencies, CoNLL, Google UPOS + UDT, HamleDT, Interset
- balances:
  details vs. simplicity (linguists vs. NLP applications)
  adaquacy for a given lang. vs. cross-ling
- started in 2014 (kick-off, guidelines v1)
- first release in 2015, new release each 6 months

UD releases

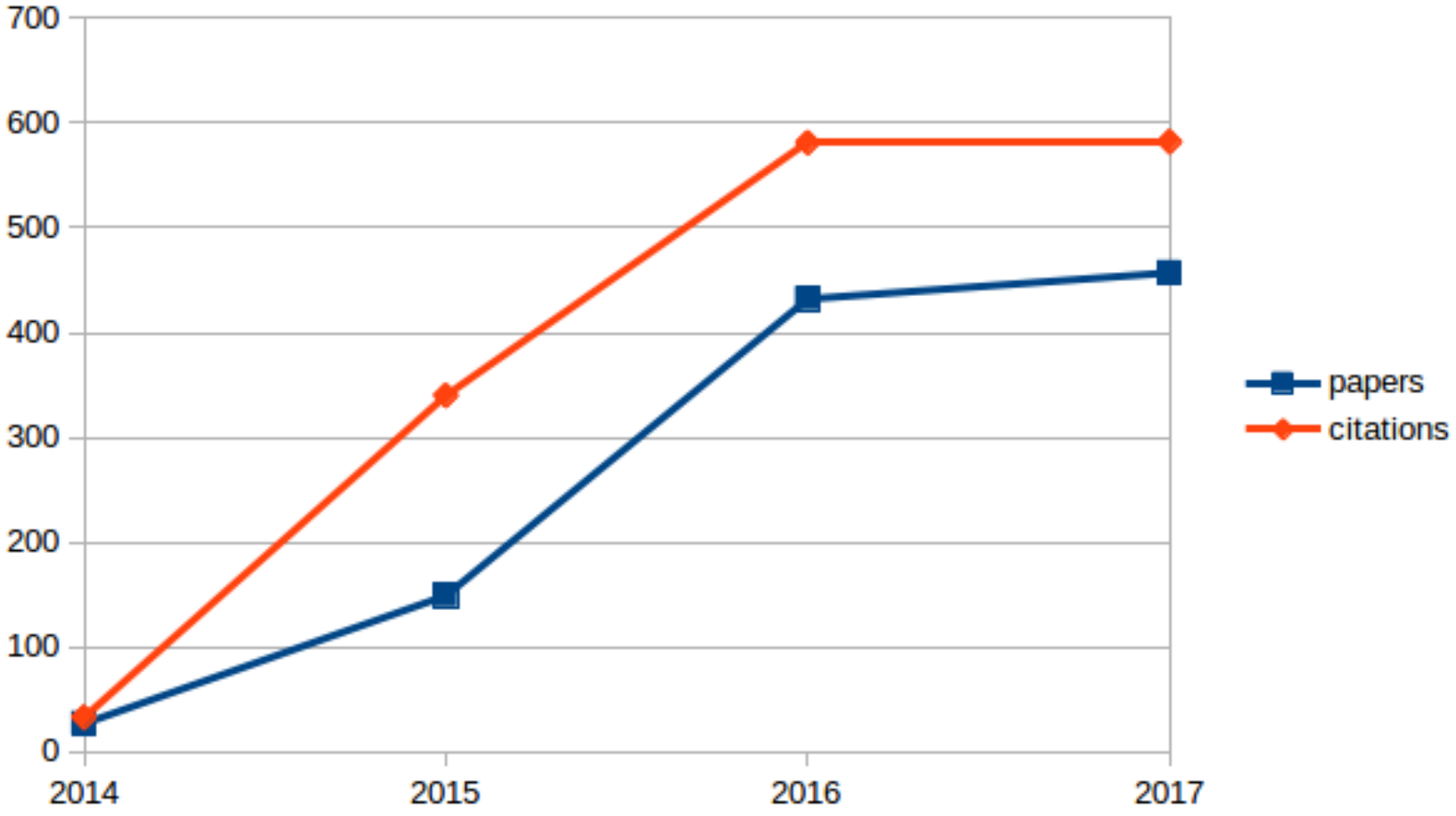
UDv2.0 released on 2017-03-01:
145 contributors, 70 treebanks, 50 langs, 12M words



(sources: `http://universaldependencies.org/`, http://lindat.cz)

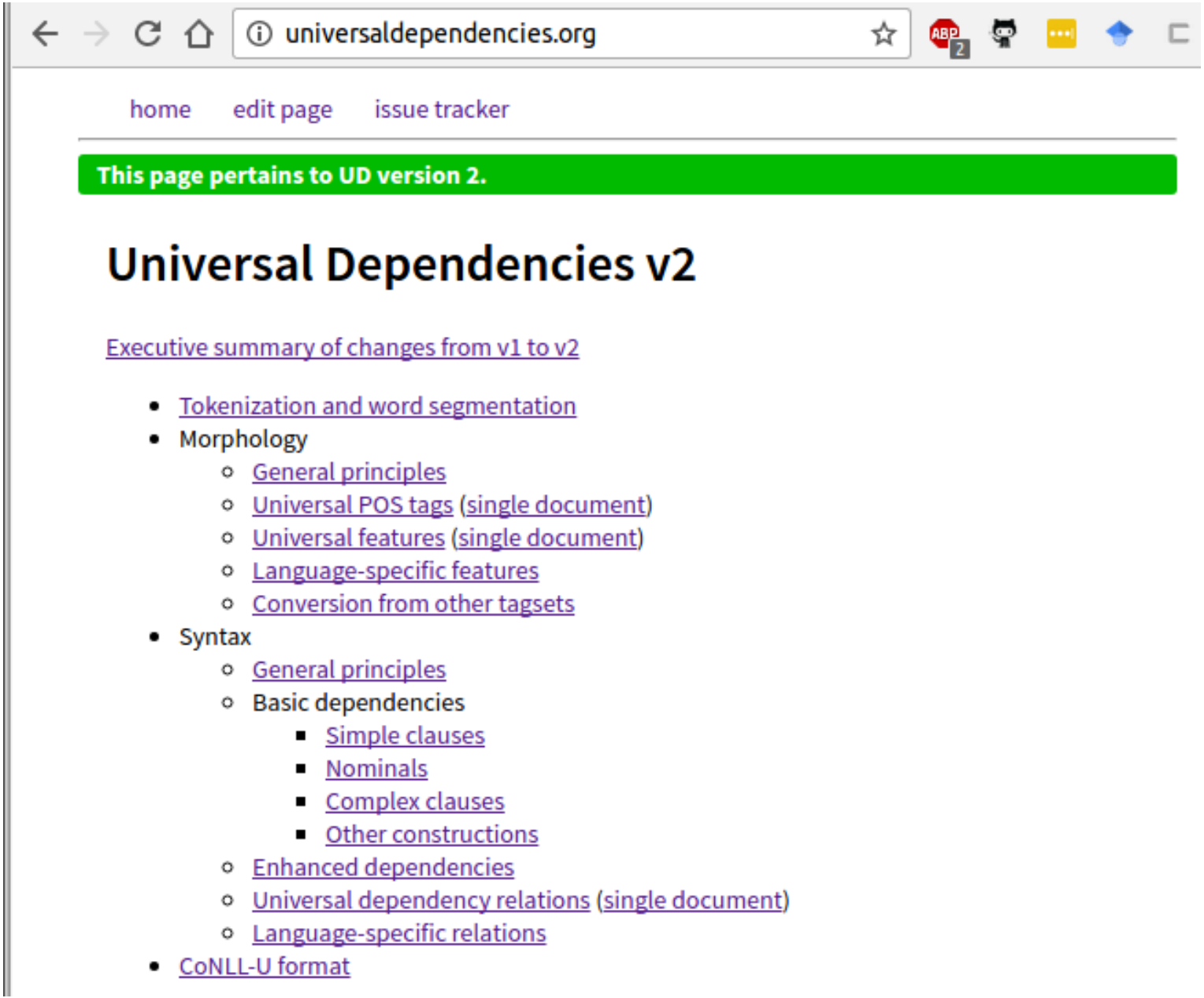
UD popularity wrt. academic papers

457 papers about Universal Dependencies, 582 citations



(source: Google Scholar, March 2017)

universaldependencies.org



Universal POS tags

Open class words	Closed class words	Other
ADJ	ADP	PUNCT
ADV	AUX	SYM
INTJ	CCONJ	X
NOUN	DET	
PROPN	NUM	
VERB	PART	
	PRON	
	SCONJ	

Universal features

Lexical features	Inflectional features	
	*Nominal**	*Verbal**
PronType	Gender	VerbForm
NumType	Animacy	Mood
Poss	Number	Tense
Reflex	Case	Aspect
Foreign	Definite	Voice
Abbr	Degree	Evident
		Polarity
		Person
		Polite

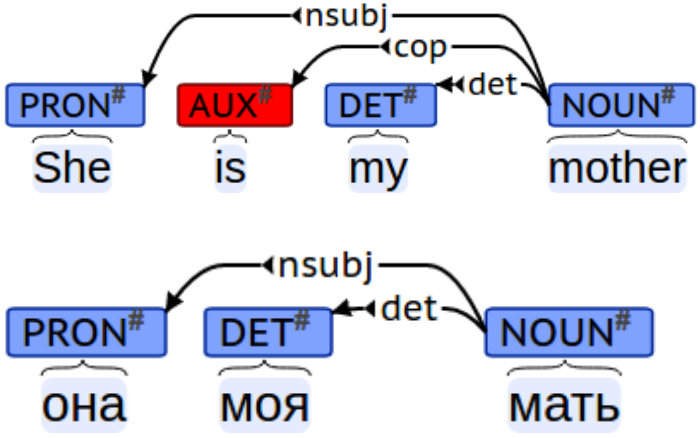
Index: A abbreviation, abessive, ablative, absolute superlative, absolutive, accusative, active, additive, adessive, admirative, adverbial participle, affirmative, allative, animate, antipassive, aorist, article, aspect, associative, **B** benefactive, **C** cardinal, case, causative case, causative voice, collective noun, collective numeral, collective pronominal, comitative, common gender, comparative case, comparative degree, complex definiteness, conditional, conjunctive, construct state, converb, count plural, counting form, **D** dative, definite, definiteness, degree of comparison, delative, demonstrative, desiderative, destinative, direct case, direct voice, directional allative, distributive case, distributive numeral, dual, **E** elative, elevated referent, emphatic, equative

Deprels (dependency relations)

	Nominals	Clauses	Modifier words	Function Words
Core arguments	nsubj obj iobj	csubj ccomp xcomp		
Non-core dependents	obl vocative expl dislocated	advcl	advmod* discourse	aux cop mark
Nominal dependents	nmod appos nummod	acl	amod	det clf case
Coordination	**MWE**	**Loose**	**Special**	**Other**
conj cc	fixed flat compound	list parataxis	orphan goeswith reparandum	punct root dep

UD v2 guidelines changes

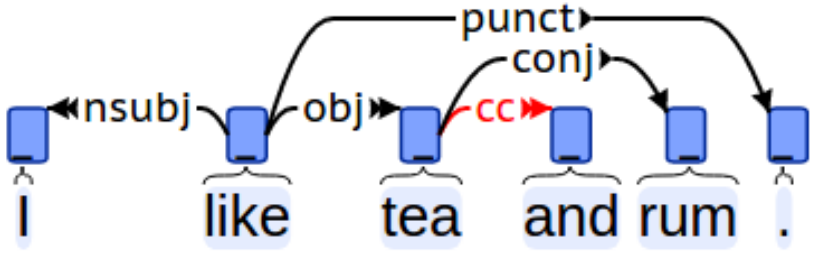
- spaces in forms and lemmas (Vietnamese, “100 000”, “i. e.”)
- few tags, deprels and features renamed (CONJ→CCONJ, dobj→obj, Negative→Polarity,. . .)
- new deprels (obl, clf), features (Polite,. . .) and values
- removed deprels (neg, auxpass)
- copula verbs tagged as AUX



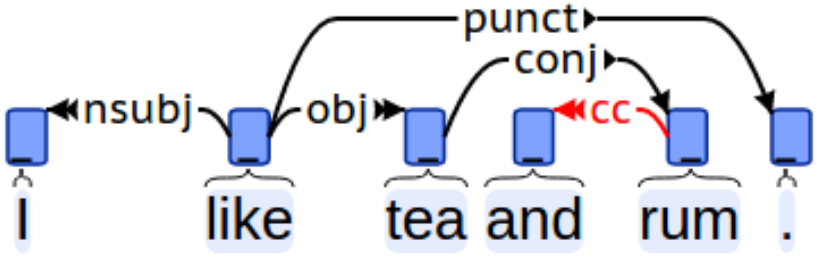
UD v2 guidelines changes

- coordinating conjunction & punctuation attached to the next conjunct (not the first conjunct)

UDv1:



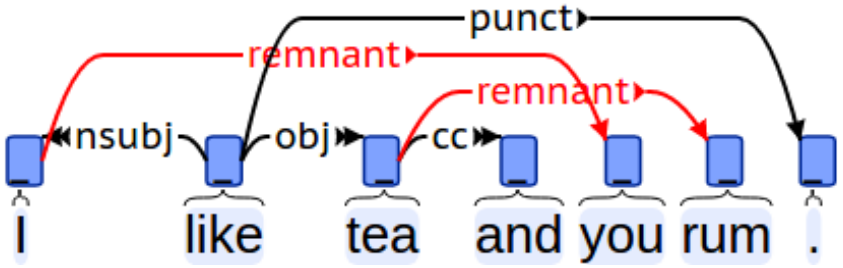
UDv2:



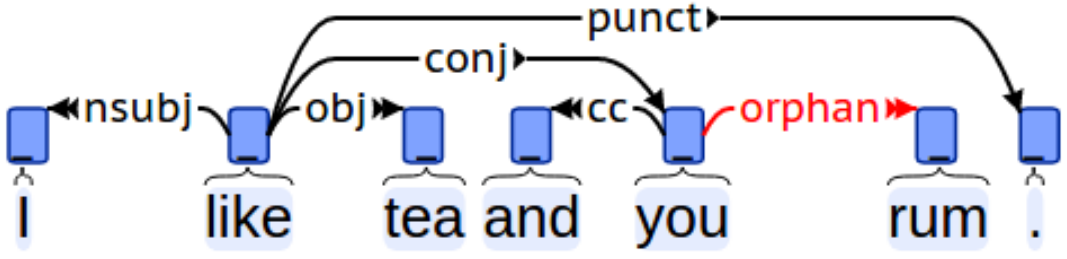
UD v2 guidelines changes

- remnant-style ellipsis → orphan-style

UDv1:



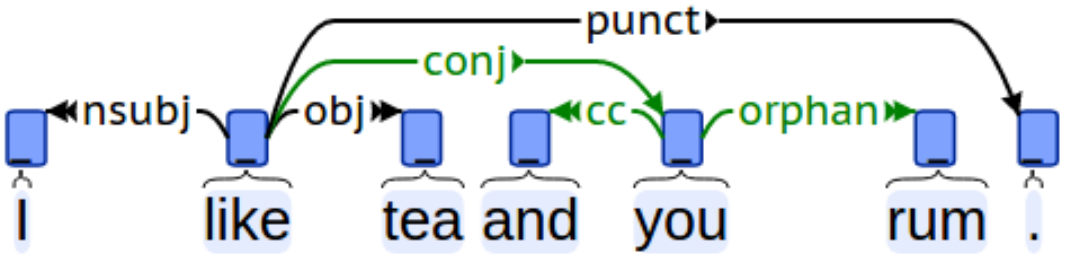
UDv2:



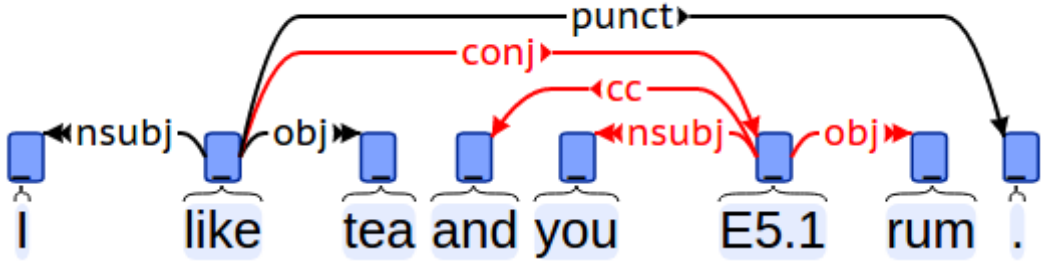
UD v2 guidelines changes

- empty nodes allowed in enhanced deps

basic:



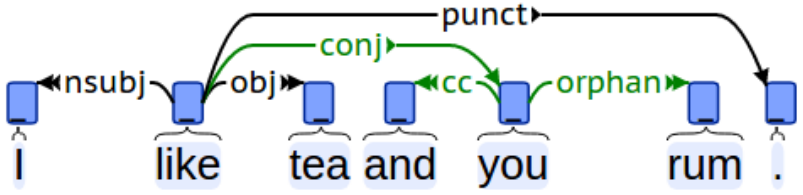
enhanced:



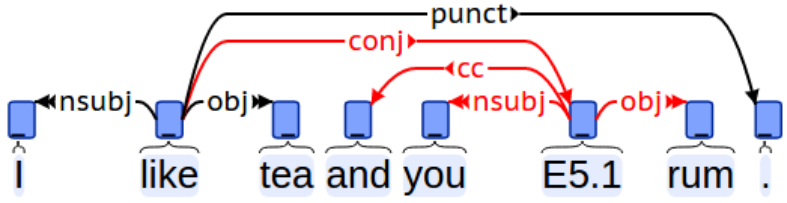
UD v2 guidelines changes

- empty nodes allowed in enhanced deps
- DEPS column with all enhanced deps (not just the extra)

basic:



enhanced:

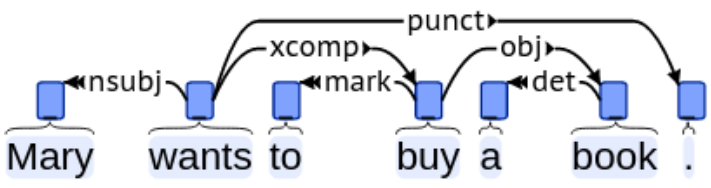



```
1    I     I     PRON  PRP  Number=Sing|Person=1|PronType=Prs  2 nsubj   2:nsubj   _
2    like  like  VERB  VBP  Mood=Ind|Tense=Pres|VerbForm=Fin  0 root    0:root    _
3    tea   tea   NOUN  NN   Number=Sing                        2 obj     2:obj     _
4    and   and   CCONJ CC   _                                  5 cc      5.1:cc    _
5    you   you   PRON  PRP  Case=Nom|Person=2|PronType=Prs     2 conj    5.1:nsubj_
5.1  _     _     VERB  VBP  Mood=Ind|Tense=Pres|VerbForm=Fin  _ _       2:conj    _
6    rum   rum   NOUN  NN   Number=Sing                        5 orphan  5.1:obj   SpaceAfter=N
7    .     .     PUNCT .    _                                  2 punct   2:punct   _
```

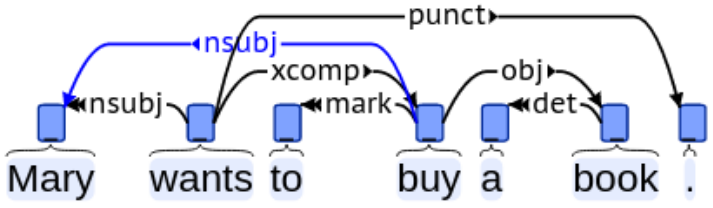
UD v2 guidelines changes

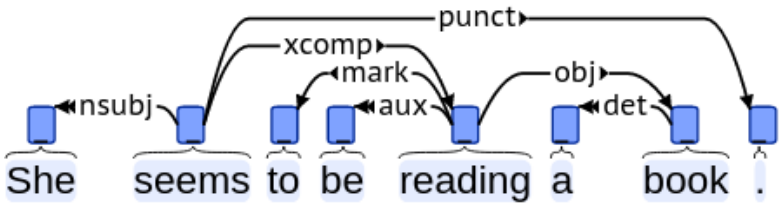
- 4 types of enhanced dependencies specified
  - controlled/raised subjects
  - ellipsis
  - propagation of conjunct
  - relative clauses
  - case information

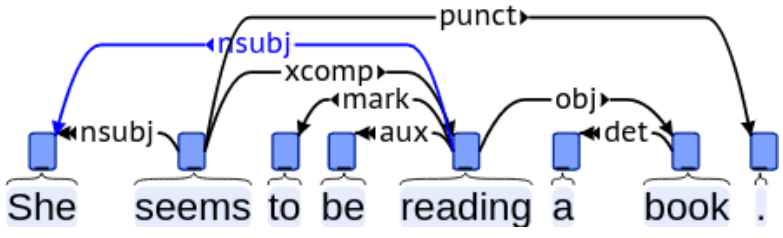
Basic



Enhanced

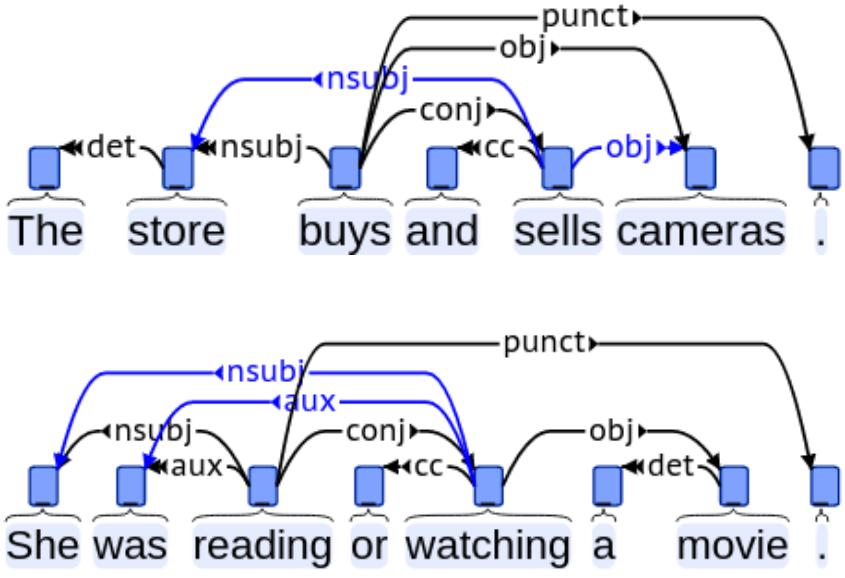






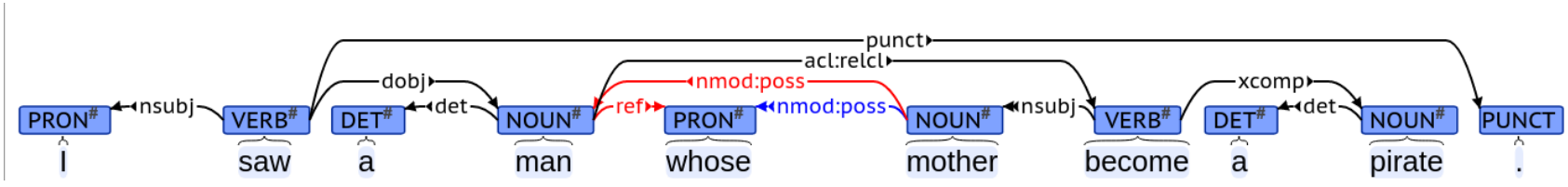
UD v2 guidelines changes

- 4 types of enhanced dependencies specified
  - controlled/raised subjects
  - ellipsis
  - propagation of conjunct
  - relative clauses
  - case information



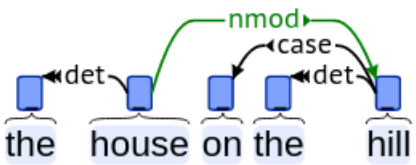
UD v2 guidelines changes

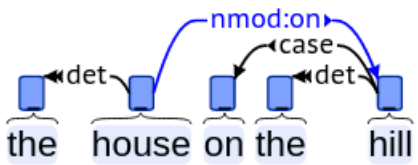
- 4 types of enhanced dependencies specified
  - controlled/raised subjects
  - ellipsis
  - propagation of conjunct
  - relative clauses
  - case information

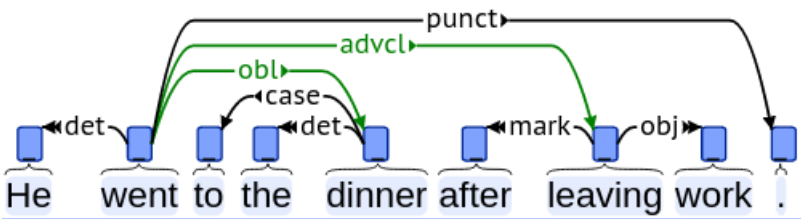


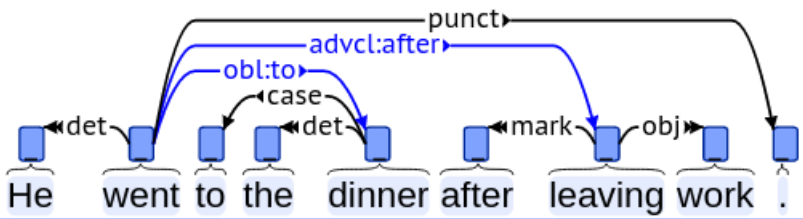
UD v2 guidelines changes

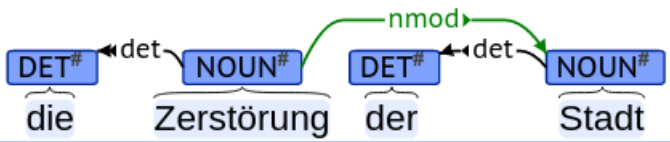
- 4 types of enhanced dependencies specified
  - controlled/raised subjects
  - ellipsis
  - propagation of conjunct
  - relative clauses
  - case information

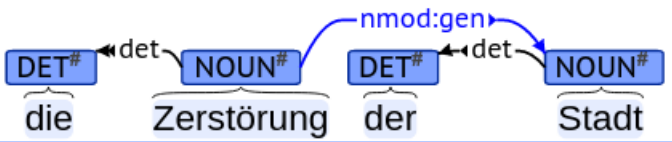










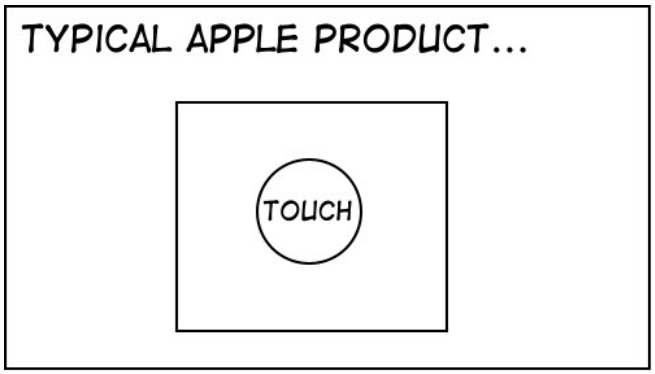


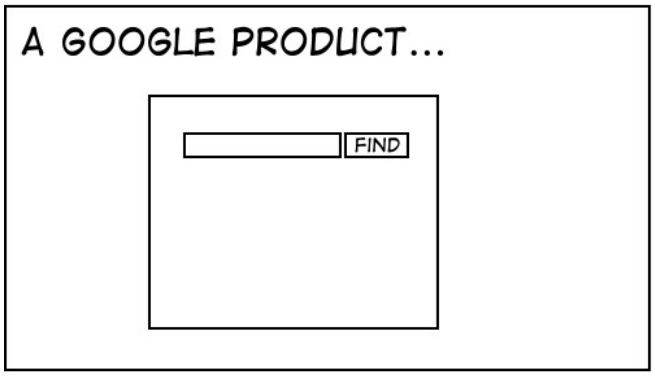
UD v2 guidelines changes

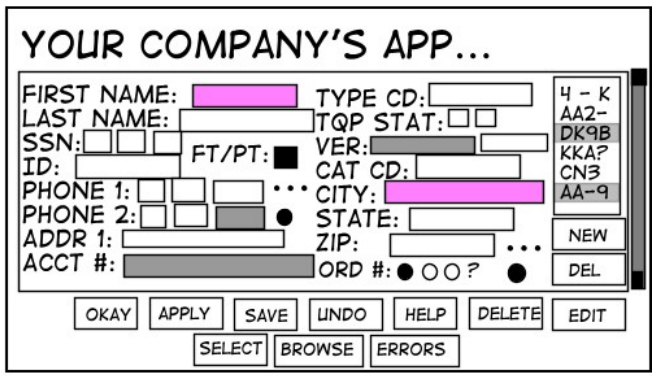
- `sent_id` and `text` comments required (and SpaceAfter=No)

```
# sent_id = 1
# text = I like tea and you rum.
1   I    I    PRON  PRP  Number=Sing|Person=1|PronType=Prs 2 nsubj  2:nsubj  _
2   like like VERB  VBP  Mood=Ind|Tense=Pres|VerbForm=Fin  0 root   0:root   _
3   tea  tea  NOUN  NN   Number=Sing                       2 obj    2:obj    _
4   and  and  CCONJ CC   _                                  5 cc     5.1:cc   _
5   you  you  PRON  PRP  Case=Nom|Person=2|PronType=Prs    2 conj   5.1:nsubj_
5.1 _    _    VERB  VBP  Mood=Ind|Tense=Pres|VerbForm=Fin _ _      2:conj   _
6   rum  rum  NOUN  NN   Number=Sing                       5 orphan 5.1:obj  SpaceAfter=N
7   .    .    PUNCT .    _                                  2 punct  2:punct  _
```

How simple is simple enough?







STUFFTHATHAPPENS.COM BY ERIC BURKE

File format: CoNLL-U (1 file, 6 lines, 218 bytes)

```
# sent_id = 1
# text = John loves Mary
1 John  John PROPN NNP Number=Sing                                  2 nsubj _ _
2 loves love VERB  VBZ Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin 0 root  _ _
3 Mary  Mary PROPN NNP Number=Sing                                  2 dobj  _ _
```

File format: Treex PML (1 file, 80 lines, 2589 bytes)

```
<?xml version="1.0" encoding="UTF-8"?>
<treex_document xmlns="http://ufal.mff.cuni.cz/pdt/pml/">
  <head>
    <schema href="treex_schema.xml" />
  </head>
  <meta/>
  <bundles>
    <LM id="1">
      <zones>
        <zone language="und">
          <sentence>John loves Mary</sentence>
          <trees>
            <a_tree id="1/und">
              <ord>0</ord>
              <children id="n2">
                <children>
                  <LM id="n1">
                    <form>John</form>
                    <lemma>John</lemma>
                    <tag>PROPN</tag>
                    <iset>
                      <pos>noun</pos>
                      <nountype>prop</nountype>
                      <number>sing</number>
                    </iset>
                    <ord>1</ord>
                    <deprel>nsubj</deprel>
                    <conll>
                      <deprel>nsubj</deprel>
                      <cpos>PROPN</cpos>
                      <pos>NNP</pos>
                      <feat>Number=Sing</feat>
                    </conll>
                  </LM>
                  <LM id="n3">
                    <form>Mary</form>
                    <lemma>Mary</lemma>
                    <tag>PROPN</tag>
                    <iset>
                      <pos>noun</pos>
                      <nountype>prop</nountype>
                      <number>sing</number>
                    </iset>
                    <ord>3</ord>
                    <deprel>dobj</deprel>
                    <conll>
                      <deprel>dobj</deprel>
                      <cpos>PROPN</cpos>
                      <pos>NNP</pos>
                      <feat>Number=Sing</feat>
                    </conll>
                  </LM>
                </children>
                <form>loves</form>
                <lemma>love</lemma>
                <tag>VERB</tag>
                <iset>
                  <pos>verb</pos>
                  <number>sing</number>
                  <person>3</person>
                  <verbform>fin</verbform>
                  <mood>ind</mood>
                  <tense>pres</tense>
                </iset>
                <ord>2</ord>
                <deprel>root</deprel>
                <conll>
                  <deprel>root</deprel>
                  <cpos>VERB</cpos>
                  <pos>VBZ</pos>
                  <feat>Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin</feat>
                </conll>
              </children>
            </a_tree>
          </trees>
        </zone>
      </zones>
    </LM>
  </bundles>
</treex_document>
```

File format: PDT PML (3 files, 90 lines, 2006 bytes)

john.w

```
<?xml version="1.0" encoding="utf-8"?>
<wdata xmlns="http://ufal.mff.cuni.cz/pdt/pml/">
  <head>
    <schema href="wdata_schema.xml"/>
  </head>
  <meta>
    <original_format>treex</original_format>
  </meta>
  <doc id="john">
    <docmeta/>
    <para>
      <w id="w-n1">
        <token>John</token>
      </w>
      <w id="w-n2">
        <token>loves</token>
      </w>
      <w id="w-n3">
        <token>Mary</token>
      </w>
    </para>
  </doc>
</wdata>
```

john.m

```
<?xml version="1.0" encoding="utf-8"?>
<mdata xmlns="http://ufal.mff.cuni.cz/pdt/pml/">
  <head>
    <schema href="mdata_schema.xml"/>
    <references>
      <reffile id="w" name="wdata" href="john.w"/>
    </references>
  </head>
  <meta>
    <lang>all</lang>
  </meta>
  <s id="m-1/und">
    <m id="m-n1">
      <w.rf>w#w-n1</w.rf>
      <form>John</form>
      <lemma>John</lemma>
      <tag>PROPN</tag>
    </m>
    <m id="m-n2">
      <w.rf>w#w-n2</w.rf>
      <form>loves</form>
      <lemma>love</lemma>
      <tag>VERB</tag>
    </m>
    <m id="m-n3">
      <w.rf>w#w-n3</w.rf>
      <form>Mary</form>
      <lemma>Mary</lemma>
      <tag>PROPN</tag>
    </m>
  </s>
</mdata>
```

(no morpho features)

john.a

```
<?xml version="1.0" encoding="utf-8"?>
<adata xmlns="http://ufal.mff.cuni.cz/pdt/pml/">
  <head>
    <schema href="adata_schema.xml"/>
    <references>
      <reffile id="m" name="mdata" href="john.m"/>
      <reffile id="w" name="wdata" href="john.w"/>
    </references>
  </head>
  <trees>
    <LM id="a-1/und">
      <s.rf>m#m-1/und</s.rf>
      <ord>0</ord>
      <children>
        <LM id="a-n2">
          <m.rf>m#m-n2</m.rf>
          <afun>Pred</afun>
          <ord>2</ord>
          <children>
            <LM id="a-n1">
              <m.rf>m#m-n1</m.rf>
              <afun>Sb</afun>
              <ord>1</ord>
            </LM>
            <LM id="a-n3">
              <m.rf>m#m-n3</m.rf>
              <afun>Obj</afun>
              <ord>3</ord>
            </LM>
          </children>
        </LM>
      </children>
    </LM>
  </trees>
</adata>
```

Prague Markup Language (PML)

Quotes from the newest documentation:

- PML is a common basis of an open family of XML-based data formats.
- PML is an on-going project in its early stage.

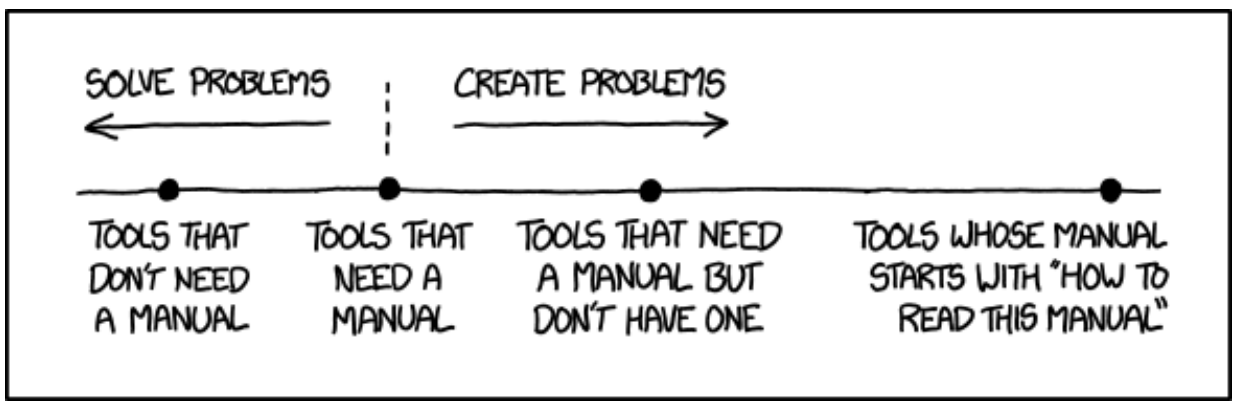
Facts:

- PML developed at ÚFAL in 2003–2006.
- PML is used in Prague-* treebanks (and lexicons), Tamil, Indonesian, Latin and Ancient Greek DT, Lithuanian DT,...
- PML has just one implementation (Perl): Fslib/TrEd/Treex::PML
- PML is difficult to understand and maintain.

UD 2.0 & CoNLL-U vs. PDT 2.0 & PML

Guidelines (number of pages) – Let's compare uncomparable...

	UD	PDT
intro	3	53
morpho	61	56
syntax	60	317
tecto	–	1287
format	7	22
schema	–	59
total	131	1794



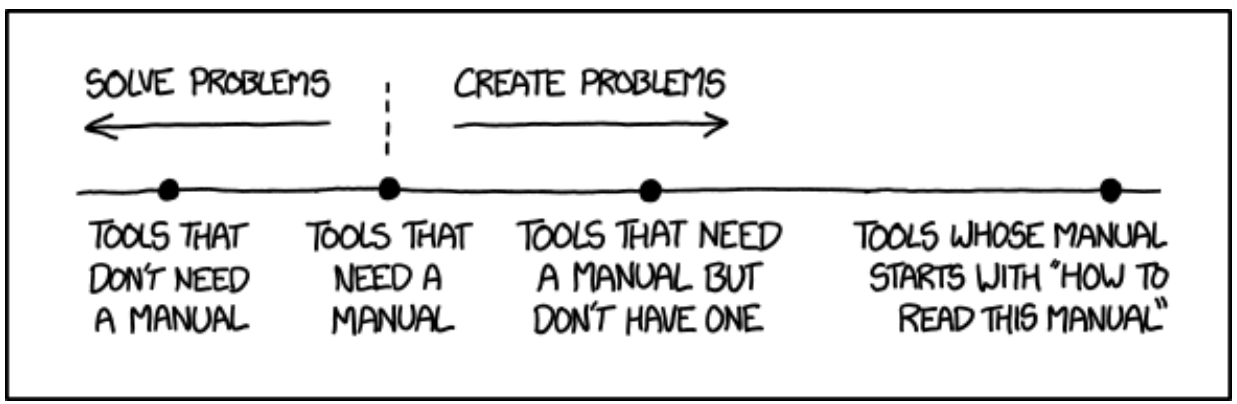
(**sources**: http://universaldependencies.org/, http://ufal.mff.cuni.cz/pdt2.0/, http://ufal.mff.cuni.cz/jazz/PML/, https://xkcd.com/1343/)

UD 2.0 & CoNLL-U vs. PDT 2.0 & PML

Guidelines (number of pages) – Let's compare uncomparable. . .

	UD	PDT
intro	3	53
morpho	61	56
syntax	60	317
tecto	–	1287
format	7	22
schema	–	59
total	131	1794

guide "to become quickly familiar with the basic ideas"



(**sources**: http://universaldependencies.org/, http://ufal.mff.cuni.cz/pdt2.0/, http://ufal.mff.cuni.cz/jazz/PML/, https://xkcd.com/1343/)

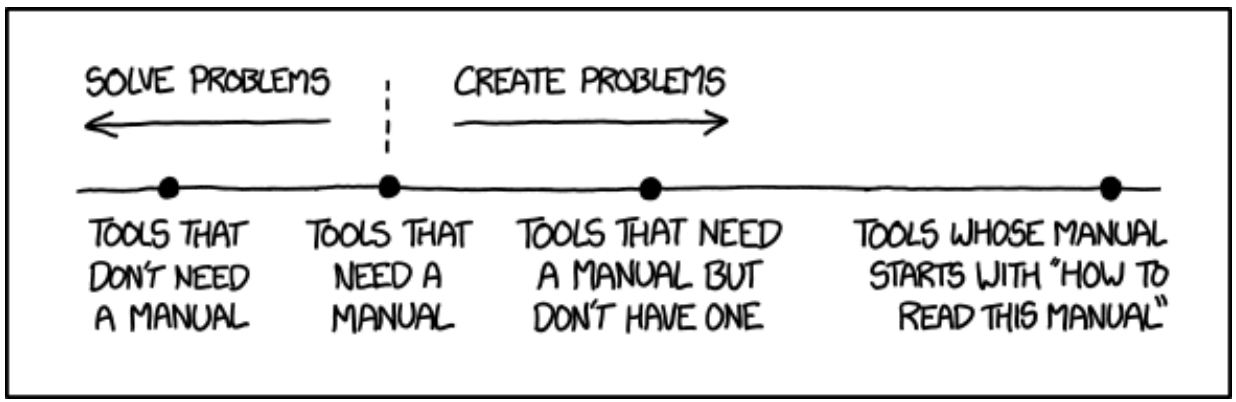
UD 2.0 & CoNLL-U vs. PDT 2.0 & PML

Guidelines (number of pages) – Let's compare uncomparable. . .

	UD	PDT
intro	3	53
morpho	61	56
syntax	60	317
tecto	–	1287
format	7	22
schema	–	59
total	131	1794

guide "to become quickly familiar with the basic ideas"

"PML also tries to retain simplicity."



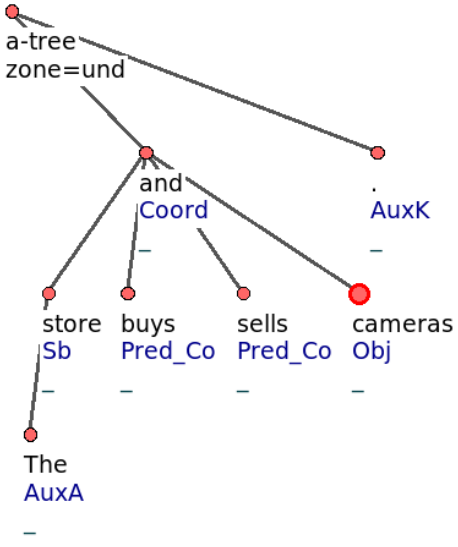
(**sources**: http://universaldependencies.org/, http://ufal.mff.cuni.cz/pdt2.0/, http://ufal.mff.cuni.cz/jazz/PML/, https://xkcd.com/1343/)

Simplicity?

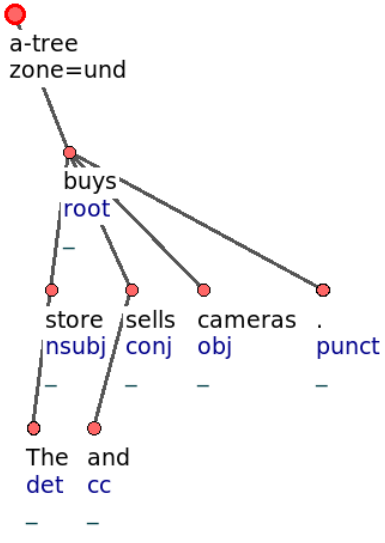
- Simplicity is not only about file size or doc pages.
- Success is not only about simplicity.
- UD/CoNLL-U approach:
  Let simple things be ultra-simple
  (at the cost of edge cases being less elegant or not possible).
  - literal undescore token in CoNLL-U
  - original text including inter-sentence spaces
  - add new type of annotation
    (cross-sentence coreference, alignment)

Simplicity: Coordinations

PDT

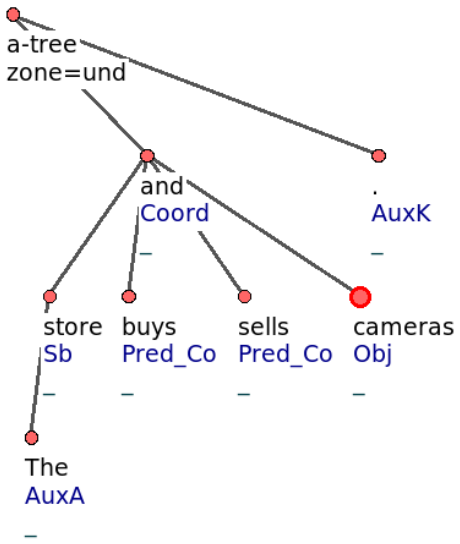


UD basic

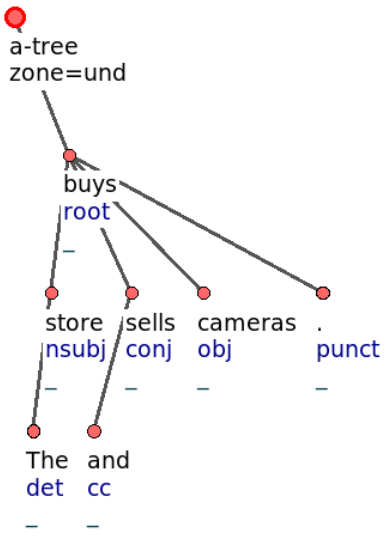


Simplicity: Coordinations

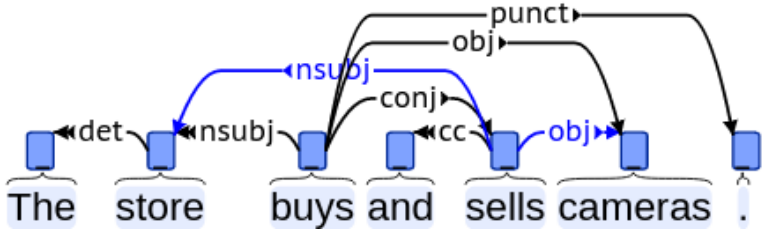
PDT



UD basic



UD enhanced

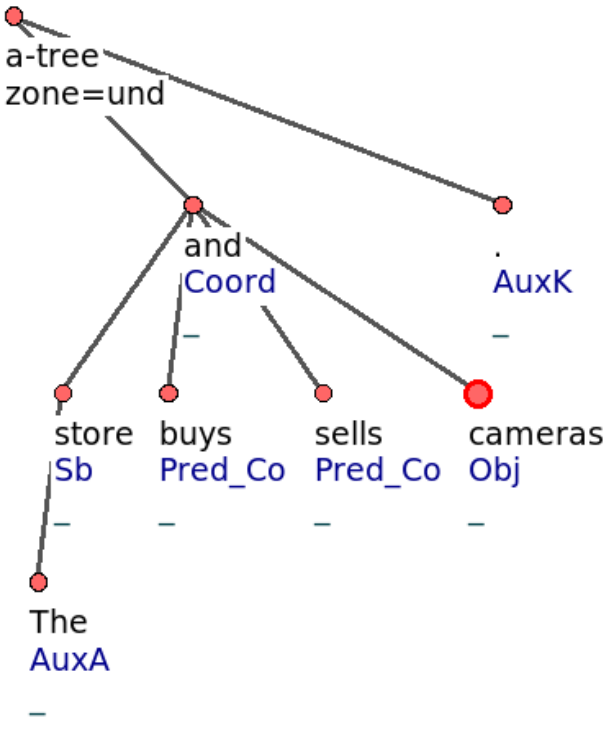


CoNLL 2017 multilingual parsing shared task

- `http://universaldependencies.org/conll17/`
- from Raw Text to Universal Dependencies
- segmentation, tokenization, labelled parsing (no morpho)
- 45 UDv2 languages plus X surprise languages
- registration deadline: April 15
- test phase: May 8–12
- organizers: ÚFAL, Google, Uppsala, Turku
- participants: 72 teams so far

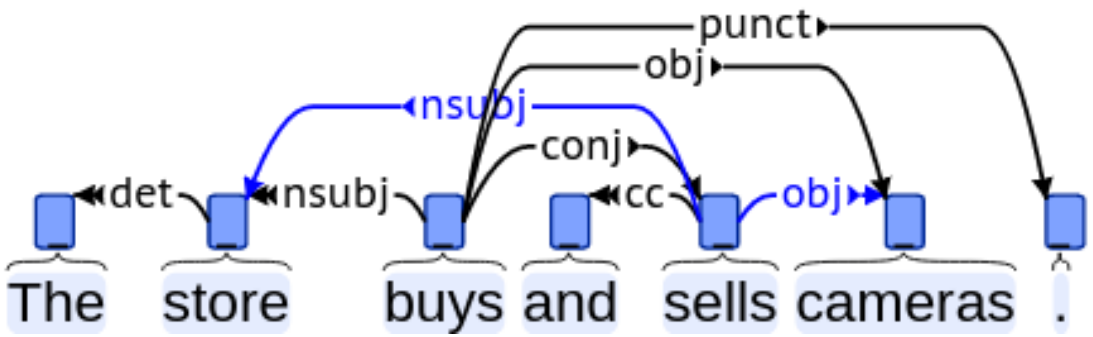
Tools for annotating dependency trees

- TrEd (+EasyTreex): powerful, customizable, Perl, old
- Brat: online/JS+Python, UD support, embeddable
- EasyTree: perhaps too simple
- GraphAnno: useful for discourse etc., Java, keyboard



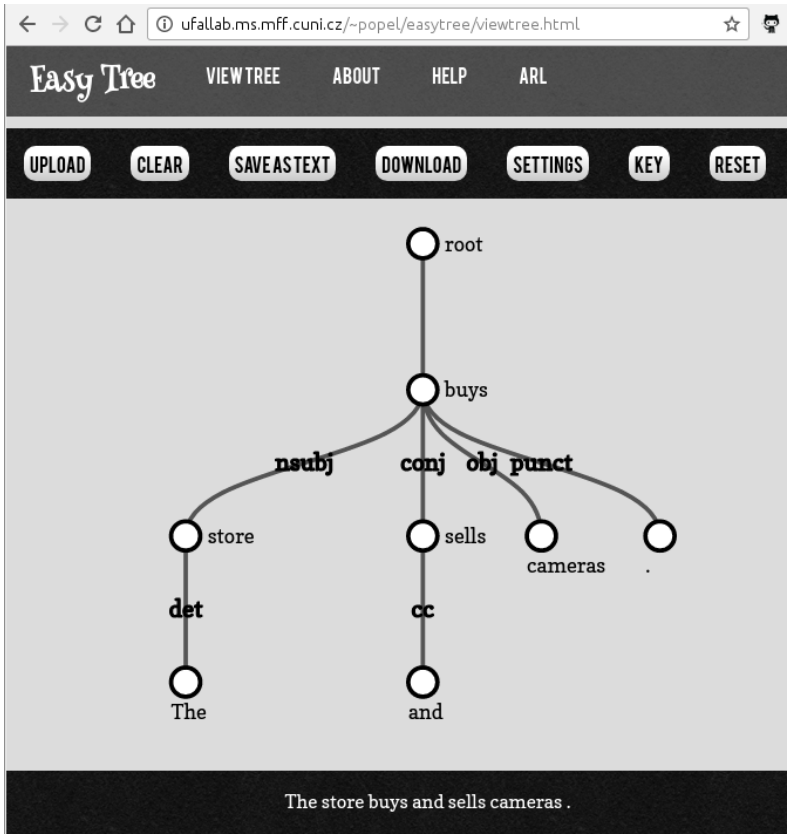
Tools for annotating dependency trees

- TrEd (+EasyTreex): powerful, customizable, Perl, old
- Brat: online/JS+Python, UD support, embeddable
- EasyTree: perhaps too simple
- GraphAnno: useful for discourse etc., Java, keyboard



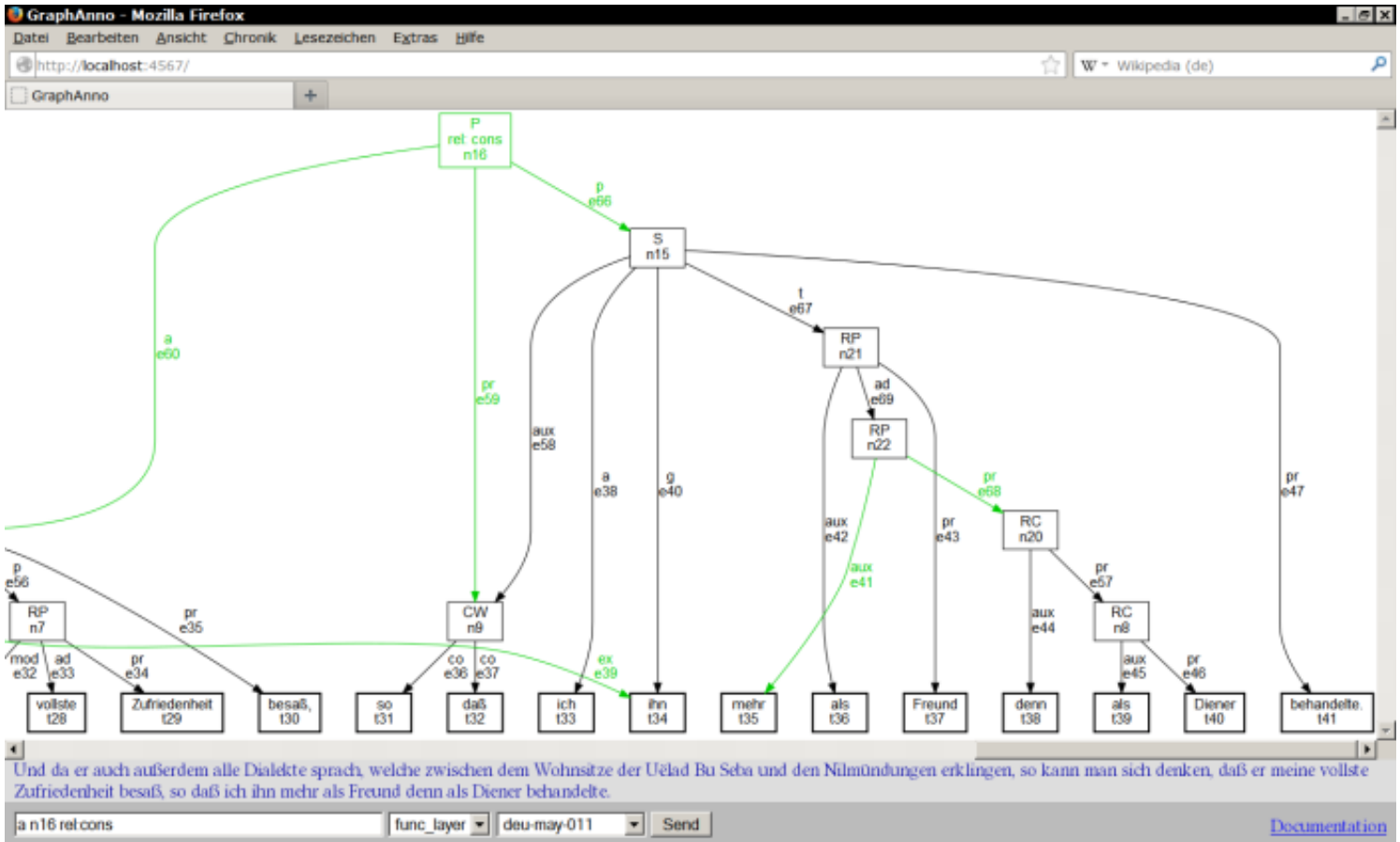
Tools for annotating dependency trees

- TrEd (+EasyTreex): powerful, customizable, Perl, old
- Brat: online/JS+Python, UD support, embeddable
- EasyTree: perhaps too simple
- GraphAnno: useful for discourse etc., Java, keyboard



Tools for annotating dependency trees

- TrEd (+EasyTreex): powerful, customizable, Perl, old
- Brat: online/JS+Python, UD support, embeddable
- EasyTree: perhaps too simple
- GraphAnno: useful for discourse etc., Java, keyboard



Automatic parsing: UDPipe

UDPipe by Milan Straka, try it online/as a webservice
`http://lindat.mff.cuni.cz/services/udpipe/`

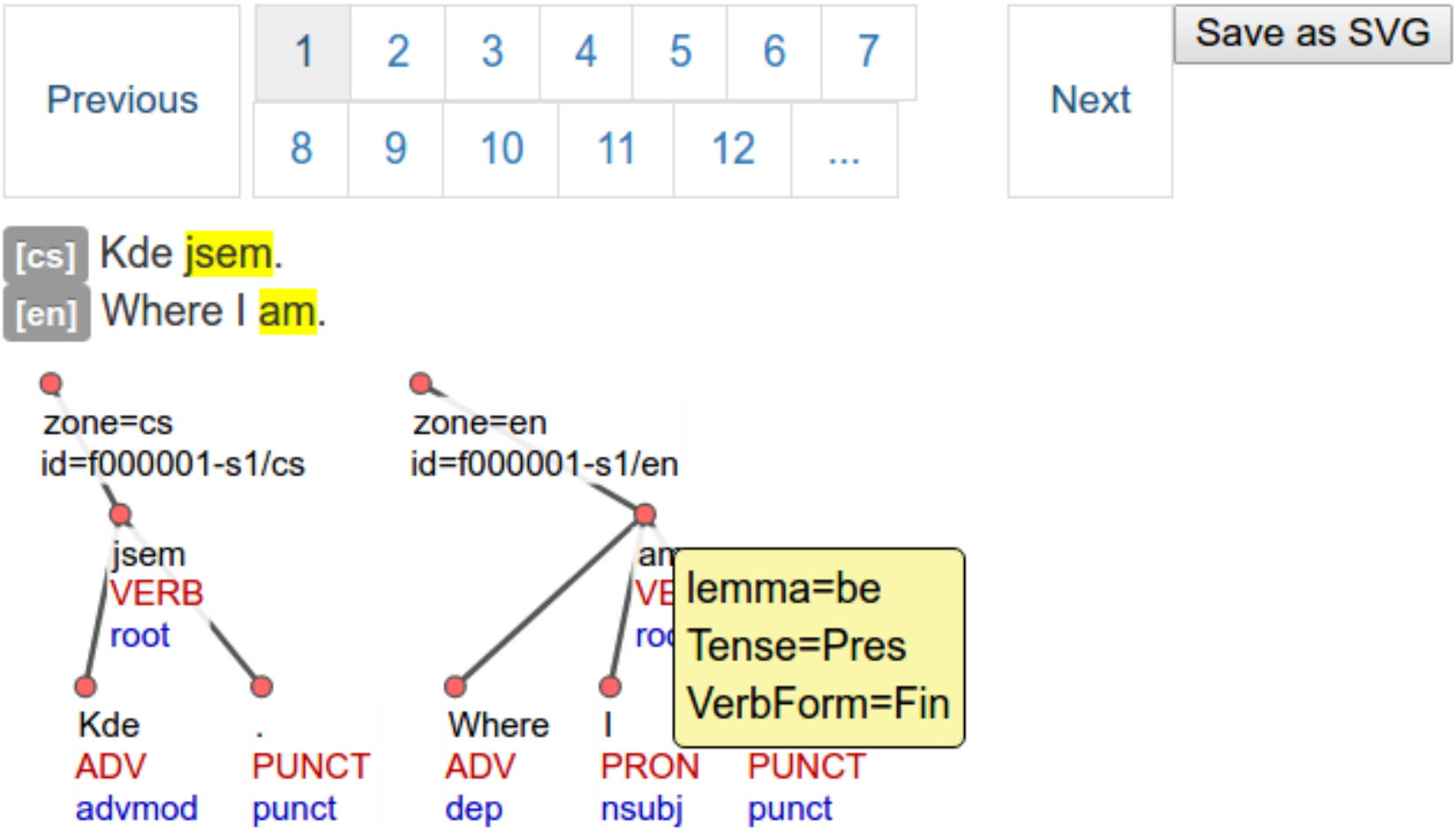
- End-to-end, batteries included:
  segment, tokenize, tag, morpho, lemma, labelled parsing
- Pretrained models for all the UD (2.0 soon) langs
- User friendly (outputs CoNLL-U, Table, SVG)
- State-of-the-art quality, ultra fast
- Open-source, easy install for Linux, OS X, Win
- Interfaces for C++, C#, Java, Perl, Python
- Easily train on your own data

Tools for viewing dependency trees

- all the editors, including UDPipe online
- PML-TQ (UDv1.2)
  `https://lindat.mff.cuni.cz/services/pmltq/`
- Udapi `https://github.com/udapi/udapi-python`

```
udapy write.Html < my.conllu > my.html
```

Tools for viewing dependency trees



Tools for viewing dependency trees

- all the editors, including UDPipe online
- PML-TQ (UDv1.2)
  `https://lindat.mff.cuni.cz/services/pmltq/`
- Udapi `https://github.com/udapi/udapi-python`

  ```
  udapy write.Html < my.conllu > my.html
  ```
  demo: `http://ufallab.ms.mff.cuni.cz/~popel/czeng1.6-sample.html`

  ```
  udapy -HA < my.conllu > my.html
  ```

Tools for viewing dependency trees

ufallab.ms.mff.cuni.cz/~popel/sv/dev-bugs.html

```
bugs = ud.MarkBugs Error Overview:
         punct-child          2
               TOTAL          2

# sent_id = sv-ud-dev-201
# text = Det snedvridna livsperspektiv som många män får i dag genom att aldrig befatta sig med 'de bal
    Det den DET DT|NEU|SIN|DEF Definite=Def|Gender=Neut|Number=Sing|PronType=Prs det _
    snedvridna snedvrida ADJ PC|PRF|UTR/NEU|SIN|DEF|NOM Case=Nom|Definite=Def|Number=Sing|Tense=P
  livsperspektiv livsperspektiv NOUN NN|NEU|SIN|IND|NOM Case=Nom|Definite=Ind|Gender=Neut|Number=
    som som PRON HP|-|-|- PronType=Rel obj _
      många många ADJ JJ|POS|UTR/NEU|PLU|IND/DEF|NOM Case=Nom|Degree=Pos|Number=Plur amod _
    män man NOUN NN|UTR|PLU|IND|NOM Case=Nom|Definite=Ind|Gender=Com|Number=Plur nsubj _
  får få VERB VB|PRS|AKT Mood=Ind|Tense=Pres|VerbForm=Fin|Voice=Act acl:relcl _
    i i ADP PP _ advmod _
    dag dag NOUN NN|UTR|SIN|IND|NOM Case=Nom|Definite=Ind|Gender=Com|Number=Sing fixed _
    genom genom ADP PP _ mark _
    att att PART IE _ mark _
    aldrig aldrig ADV AB Polarity=Neg advmod _
  befatta befatta VERB VB|INF|AKT VerbForm=Inf|Voice=Act advcl _
    sig sig PRON PN|UTR/NEU|SIN/PLU|DEF|OBJ Case=Acc|Definite=Def|PronType=Prs obj _
      med med ADP PP _ case _
      ' ' PUNCT PAD _ punct SpaceAfter=No
      de en DET DT|UTR/NEU|PLU|DEF Definite=Def|Number=Plur|PronType=Art det _
      bakre bakre ADJ JJ|KOM|UTR/NEU|SIN/PLU|IND/DEF|NOM Case=Nom|Degree=Cmp amod _
    linjerna linje NOUN NN|UTR|PLU|DEF|NOM Case=Nom|Definite=Def|Gender=Com|Number=Plur obl S
      ' ' PUNCT PAD _ punct _
      ( ( PUNCT PAD _ punct SpaceAfter=No
        ! ! PUNCT MAD _ punct Bug=punct-child|SpaceAfter=No
        ) ) PUNCT PAD _ punct Bug=punct-child
  är vara AUX VB|PRS|AKT Mood=Ind|Tense=Pres|VerbForm=Fin|Voice=Act cop _
  förvisso förvisso ADV AB _ advmod _
```

Tools for viewing dependency trees

- all the editors, including UDPipe online
- PML-TQ (UDv1.2)
  `https://lindat.mff.cuni.cz/services/pmltq/`
- Udapi `https://github.com/udapi/udapi-python`

  ```
  udapy write.Html < my.conllu > my.html
  ```
  demo: `http://ufallab.ms.mff.cuni.cz/~popel/czeng1.6-sample.html`

  ```
  udapy -HA < my.conllu > my.html
  ```

  ```
  udapy -T < my.conllu | less -R
  ```

Tools for viewing dependency trees

```
# sent_id = 1
# text = Corriere Sport da pagina 23 a pagina 26
Corriere PROPN root
  Sport PROPN name
    da ADP case
  pagina NOUN nmod
    23 NUM nummod
    a ADP case
  pagina NOUN nmod
    26 NUM nummod

# sent_id = 2
# text = I tre avevano da poco lasciato la cima e stavano cominciando la discesa.
    I DET det
  tre NUM nsubj
  avevano AUX aux
    da ADP case
  poco ADV advmod
lasciato VERB root
    la DET det
  cima NOUN dobj
  e CONJ cc
    stavano AUX aux
  cominciando VERB conj
      la DET det
    discesa NOUN dobj
  . PUNCT punct
```

Tools for viewing dependency trees

- all the editors, including UDPipe online
- PML-TQ (UDv1.2)
  `https://lindat.mff.cuni.cz/services/pmltq/`
- Udapi `https://github.com/udapi/udapi-python`

  ```
  udapy write.Html < my.conllu > my.html
  ```
  demo: `http://ufallab.ms.mff.cuni.cz/~popel/czeng1.6-sample.html`

  ```
  udapy -HA < my.conllu > my.html
  ```

  ```
  udapy -T < my.conllu | less -R
  ```
  T=text, H=html, A=all attributes, N=no color, M=marked

Udapi

- `http://udapi.github.io/`
- API and multi-language framework for processing UD
- Allows both fast prototyping and full applications
- Both command-line tool (`udapy`) and library
- Modularity, reusability, cooperation

Udapi use cases

- format conversions (CoNLL-U, SDParse, PML, VISL-cg, TikZ)
- `ud.Convert1to2` transformations from UD v1 to v2
  used for Bulgarian, Romanian, Galician, Russian, Irish, ...
- `ud.MarkBugs` validity tests
- `ud.SetSpaceAfter`, `ud.SetSpaceAfterFromText`
- `util.Eval`, `util.Filter`, `util.Wc`
- automatic parsing (via UDPipe), evaluation,...

Hands-on tutorial

`http://udapi.github.io/tutorial/`

Treex vs. Udapi

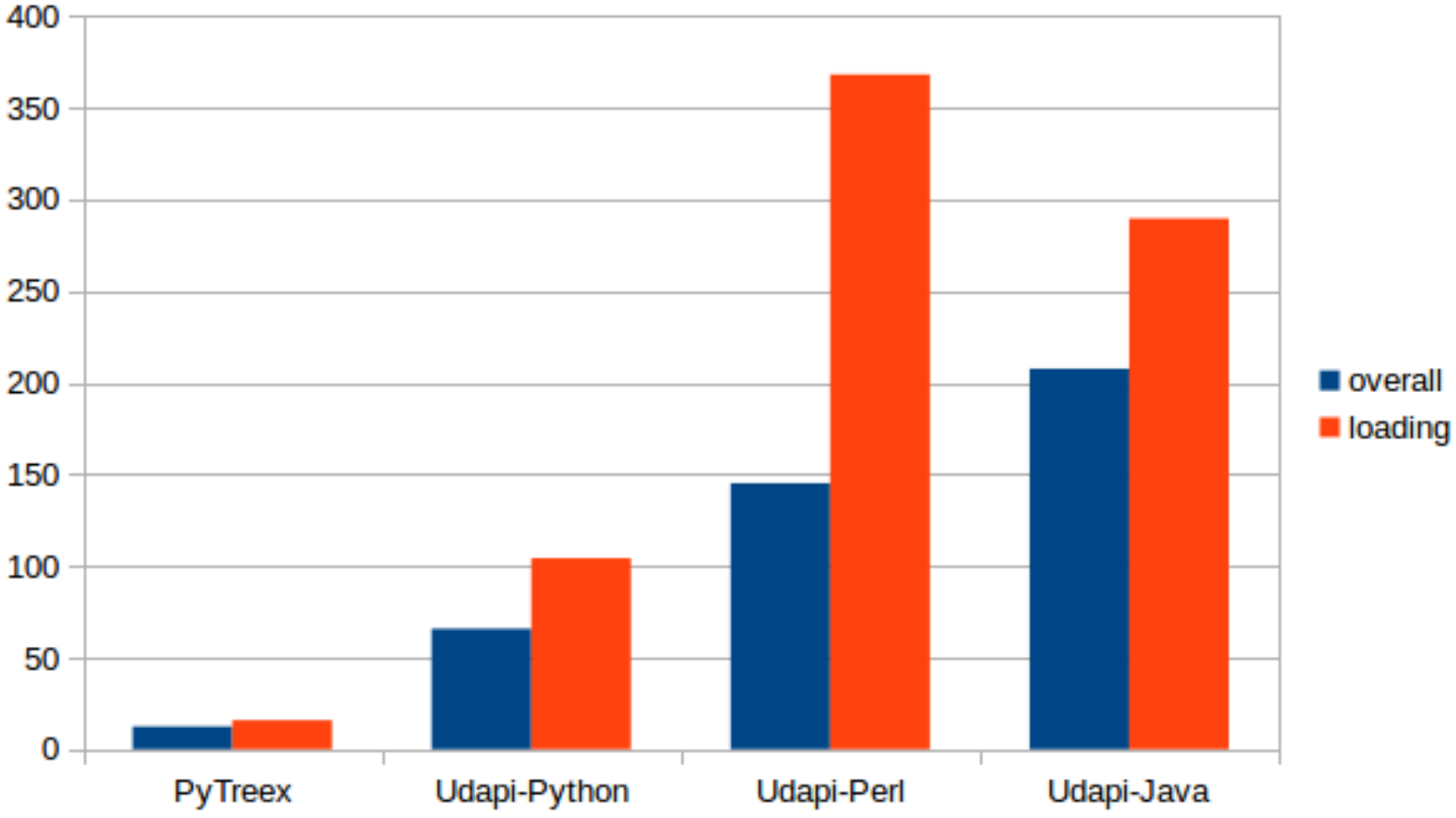
Treex (2005)

- Perl
- over 10 ÚFAL developers
- multilingual in the end
- Prague-style dependencies
- tectogrammatical layer
- PML / XML / `*.treex`
- weak UD support
- focus: TectoMT
- `https://github.com/ufal/treex`

Udapi (2016)

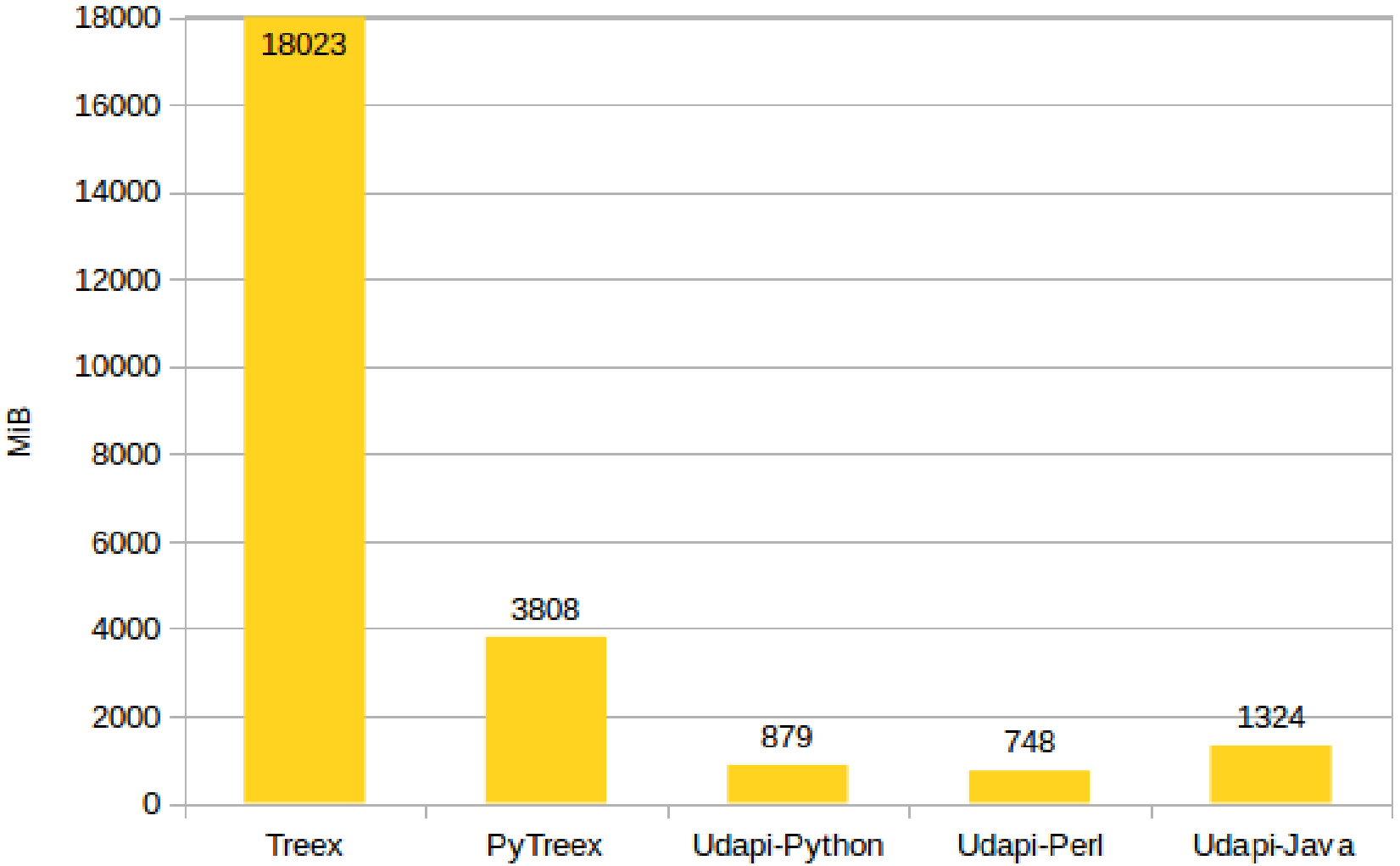
- **Python**, Perl, Java
- MP+ZŽ+M.Vojtek
- multilingual from the start
- Universal dependencies
- no layers (but zones)
- CoNLL-U
- MWT, empty, enhanced
- focus: speed & simplicity
- `https://github.com/udapi/`

Benchmark: Speed-up relative to Treex



(source: `https://github.com/martinpopel/newtreex`)

Benchmark: Memory (MB)



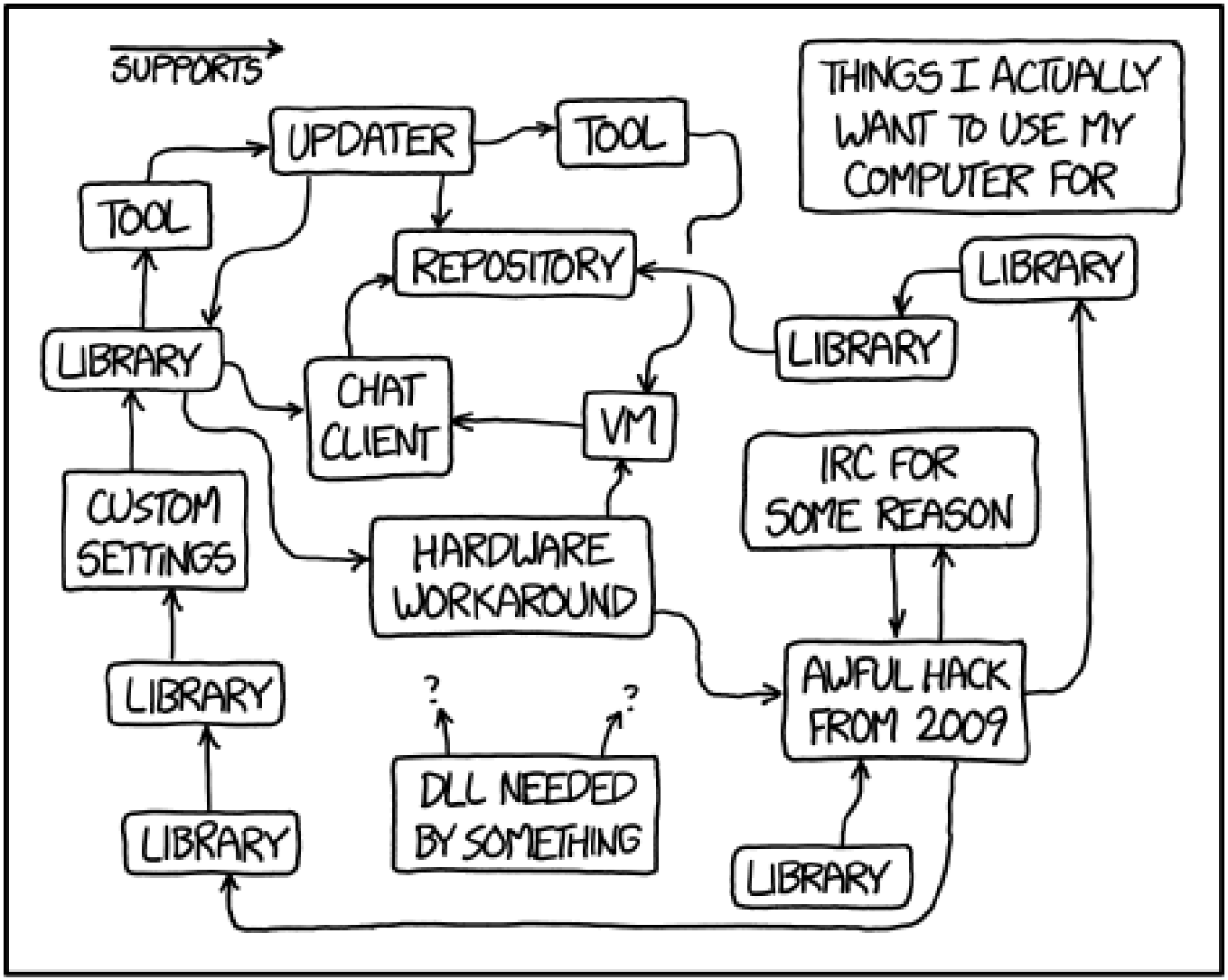
(source: `https://github.com/martinpopel/newtreex`)

`cs-ud-train-l.conllu`: 68 MiB, 41k sentences, 0.8 MWords

Algorithmic challenges

- data structure for globally-ordered rooted trees
  `node.descendants` ... ordered
  `node.shift_before_node()`
- efficient loading&saving of CoNLL-U files
  linear-time checking of cycles
  lazy deserialization of FEATS and MISC
- `write.TextModeTrees` for non-projective trees
  minimize crossings and/or depth
- align raw sentence with gold-annotated words
  add `SpaceAfter=No` and `goeswith`
  create multi-word tokens (minimal)
  mark typos etc. in MISC

Summary



https://www.xkcd.com/1579/