Querying Diverse Treebanks in a Uniform Way

Jan Štěpánek, Petr Pajas
Charles University in Prague, MFF ÚFAL

- What is "syntax"?
- Different names of categories and their values
- Various data formats
- Different tree encoding (by structure × by reference)

Treebank	Total num. of nodes	max rank	median rank	max tree	median tree	max depth	max breadth	nodes / terminals
PDT	1.59M	85	3	195	12	24	85	--
Tiger	0.95M	17	2	237	7	23	53	1.55
WSJ	2.28M	51	3	441	10	37	159	1.82
Atis	0.01M	8	2	81	10	16	17	2.13
Brown	0.92M	24	2	347	14	36	53	1.89
SWBD	2.73M	26	6	272	63	37	54	1.91
Chinese (Penn)	1.86M	64	2	558	4	30	169	2.18
Arabic	0.36M	25	2	602	173	52	73	2.16
Catalan	0.4M	37	9	215	23	24	56	--
Chinese (CoNLL)	0.63M	35	3	243	30	20	114	--
Spanish	0.44M	62	2	150	17	28	64	--

Tiger 1.0 · CoNLL-2009 Shared Task · Penn Arabic Treebank 2 · Penn CU Chinese Treebank 6.0 · PTB 3

Conversion to PML

Conversion to SQL

PDT 2.0

PML-TQ Querying

Prague Markup Language

PML Schema can define the following types:

- **Atomic**: a string, its value can further be restricted to a specific format (e.g. any, integer, date...)
- **Enumerated**: atomic type with a given set of possible values.
- **Structure**: set of attribute-value pairs.
- **List**: ordered or unordered list of constructs of one type.
- **Alternative**: similar to unordered list, but with different semantics.
- **Sequence**: similar to ordered list, but allowing members with diverse types and supporting mixed content.

PML-TQ

- selecting all **occurrences** of nodes from the treebanks with given properties and in given relations w.r.t. the **tree topology, cross-referencing, surface ordering**, etc.
- bounded or unbounded **iteration** (i.e. transitive closure) of relations
- **multi-layered** or aligned treebanks with structured attribute values
- **quantified** or **negated** subqueries
- **referencing** among nodes
- natural **textual** and **graphical** representation of the query (the structure of the query corresponds to the structure of the matched subtree)
- sublanguage for postprocessing and **generating reports** (filtering, grouping, aggregating, and sorting)
- support for **regular expressions**, basic **arithmetic** and string operations

Word-order typology (German CoNLL)

```
node $p := [ substr(pos,0,1) = 'V',
  ? node $ch := [
     deprel in {'SB','OA','OC','OA2','OP'} ] ];
>> give $p.xml:id,
      if($p=$ch,
        if($p.deprel = 'ROOT','V','v'),
        substr($ch.deprel,0,1)),
      $ch.order
>> give distinct $1,
      concat($2,'' over $1 sort by $3)
>> give
      substitute($2,'([OS])\\1+','\\1','g')
>> filter ($1 ~ 'O' and $1 ~ 'S')
>> for $1 give $1,count() sort by $2 desc
```

Main clause	Num. of occurences	Dependent clause	Num. of occurences
SVO	11267	SOv	7556
VSO	7111	SvO	2273
OVS	2209	vSO	1113
VOS	625	OSv	606
SOV	110	OvS	109
OVSO	91	vOS	64
VOSO	64	SOvO	37
OVOS	31	OSOv	34

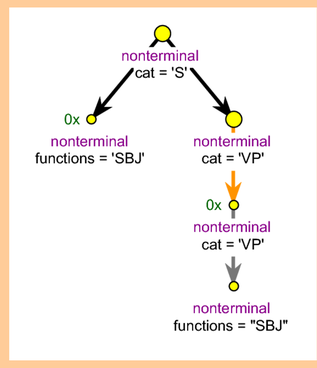
Grammar extraction (constituent trees)

```
nonterminal $p := [ * $ch := [ ] ]
>> give $p, $p.cat,
     first_defined($ch.cat,$ch.pos),
     lbrothers($ch)
>> give $2 & " -> "
     & concat($3," " over $1 sort by $4)
>> for $1 give count(),$1
     sort by $1 desc
```

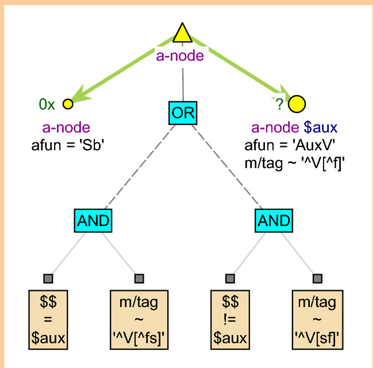
```
189856     PP → IN NP
128140      S → NP VP
 87402     NP → NP PP
 72106     NP → DT NN
 65508      S → NP VP .
 45995     NP → -NONE-
 36078     NP → DT JJ NN
 31916     VP → TO VP
 28796     NP → NNP NNP
 23272   SBAR → IN S
    ...
```

PTB

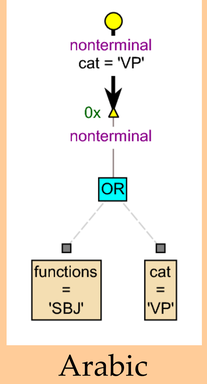
Verb clause without a subject



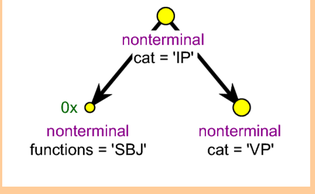
PTB



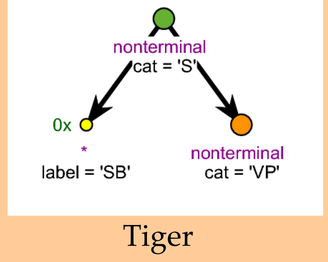
PDT



Arabic

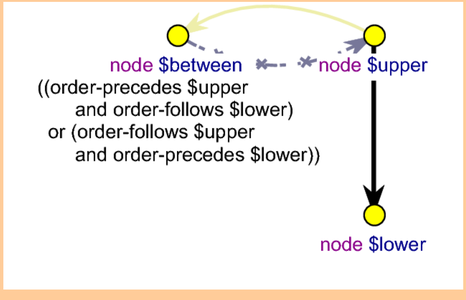


Chinese CU

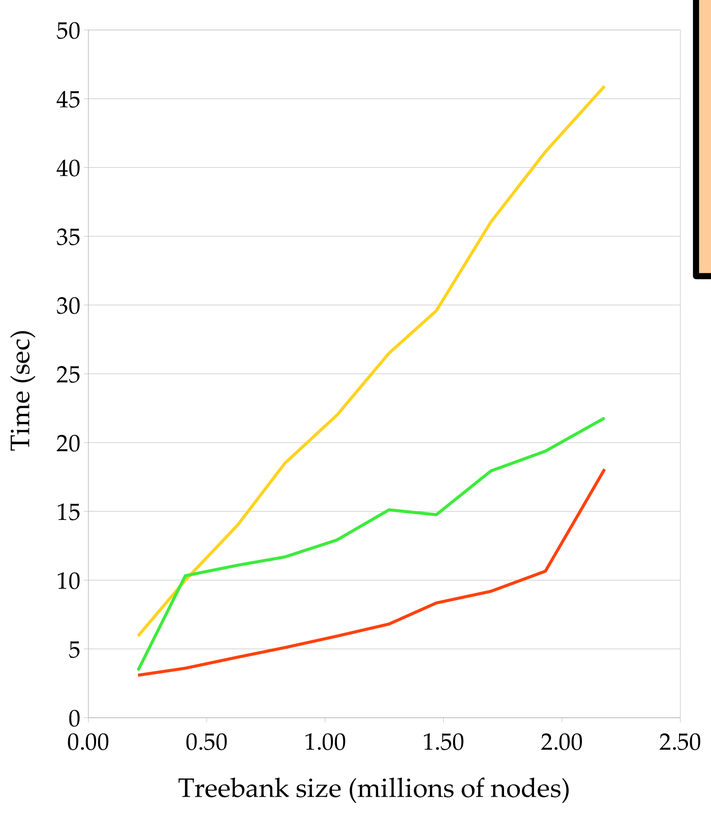


Tiger

Non-projective edges



Treebank	NPE/edge	NPE/tree	NPTrees
German CoNLL	2.33%	41.99%	28.10%
PDT 2.0	1.88%	32.14%	22.98%
English CoNLL	0.39%	9.48%	7.63%



LREC 2010, Malta. Supported by grant 406/10/P193 of Grant Agency of Science of the Czech Republic and grant FP7-ICT-2007-3-231720 (EuroMatrix Plus, 7E09003 in the Czech Rep.) of the 7th Framework Programme of the EU.