# Newer DELPH-IN Tools

9th DELPH-IN Summit
31 July 2013

# Stag

- Workflow management for corpus tasks (parsing, generation, etc.)
- Provenance tracking
- Store and use arbitrary information
- Custom commands
- Useful for those applying parser/generator output to other tasks
- Available at: https://github.com/goodmami/stag
- Usage:

```
$ stag [-c cfg] [-s src_gram] [-i infile] [-o outdir] tasks
```

# Example:

```
$ head -n1 tc-006.en
1090000     {"sentence": "Does he have anything to do with
the campaign?"}
$ stag -s ~/erg/erg.dat -i tc-006.en -o output/ s j
$ head -n2 tc-006.en.s
1090000  s:0   {"derivation": "(...)", "mrs": "[...]"}
1090000  s:1   {"derivation": "(...)", "mrs": "[...]"}
$ head tc-006.en.s.j
1090000  s:0|j:0  {"sentence": "Does he have anything to do
with the campaign?"}
1090000  s:0|j:1  {"sentence": "He does have anything to do
with the campaign?"}
...
1090000  s:1|j:0  {"sentence": "Does he have anything to do
with the campaign?"}
```

# Requires:

- Python3
- Streamable data (currently only .stag format)

# Configure (.stagrc):

```
"commands": {
 "s": {
   "description": "Parse using the source grammar.",
   "type": "ace-parse",
   "arguments": ["-g", "{source_grammar}"]
 }, ...
"commandtypes": {
 "ace-parse": {
   "description":"Use ACE to parse sentences.",
   "input_key":"sentence",
   "executable":"/usr/bin/ace",
   "arguments":["-n", "{nsolutions}"],
   "interface":"ace",
   "task":"parse"
 }, ...
```

# pyDelphin (updates)

- Python libs for working with DELPH-IN data
- Earlier features:
  - Reading/writing [incr tsdb()] profiles
  - MRS isomorphism checking
- Additions since last year:
  - MRX reading/writing
  - DMRX reading/writing
  - Support for all Lnk types (charspan, chartspan, edge, tokens)---at least for SimpleMRS
  - Pred-string parsing
- Available at: https://github.com/goodmami/pydelphin

- Useful for Python programmers or those wanting lightweight solutions for simple tasks
- Command-line scripts for [incr tsdb()] profile updating and MRS format conversion:

```
$ itsdb.py update -r new-relations profile-path

$ mrs.py convert --from format1 --to format2 file
```

# CONVERTER

- converts ERG derivation tree to bilexical syntactic dependencies

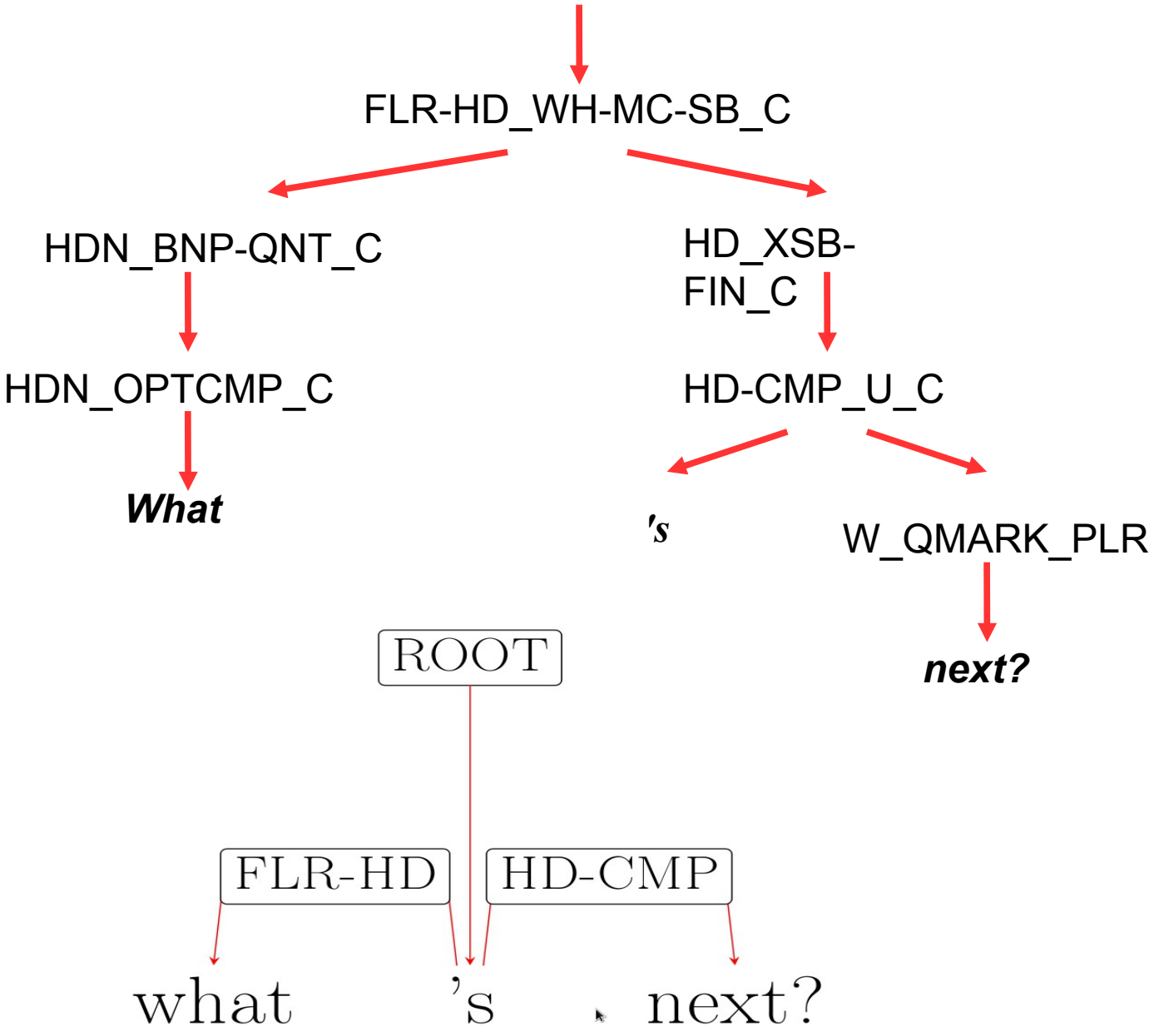
- converts ERG Elementary Dependency Structures (EDS) to bilexical semantic dependencies

- Use case: dependency parsing with ERG

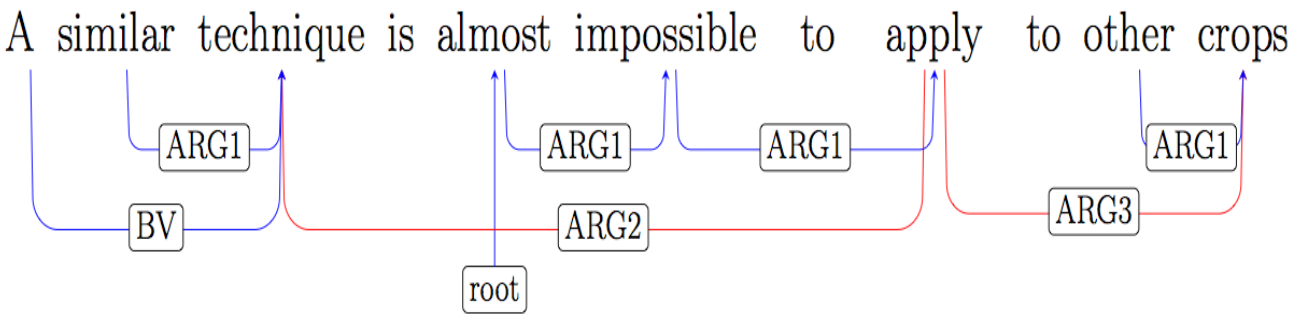
Location: **${LOGONROOT}/uio/dtm/converter.py**

Usage: ${LOGONROOT}/bin/dtm **--grammar** <gr> **--data** <data> **--tok** [erg|ptb] **--dt** <dt> **--dm** <dm> **--dtm** <dtm> **--tex** <tex>

# DELPH-IN Syntactic Derivation Tree (DT)



# DELPH-IN MRS-derived Dependencies (DM)




{ e12
_1:_a_q(BV x6)
e9:_similar_a_to(ARG1 x6)
**x6:_technique_n_1**
e12:_almost_a_1(ARG1 e3)
e3:_impossible_a_for(ARG1 e18)
**e18:_apply_v_to(ARG2 x6, ARG3 x19)**
_2:udef_q(BV x19)
e25:_other_a_1(ARG1 x19)
**x19:_crop_n_1**
…
}