PyDelphin 1.0.0

July 15, 2019

1 PyDelphin 1.0.0

"The one where everything breaks"

1.1 Packaging

- Package hierarchy is flatter (e.g., `delphin.interfaces.ace` -> `delphin.ace`)
- `delphin` is a namespace package
- ... so is `delphin.codecs`
- This allows for plugins:
  - https://github.com/delph-in/delphin.highlight
  - https://github.com/delph-in/delphin-latex
  - https://github.com/delph-in/delphin.redwoods
- And makes it possible to spin off modules as separately maintained plugins

1.2 Documentation

- API documentation is comprehensive (please file a bug otherwise)
- Includes several guides for more general advice
- Creation of some new modules (`delphin.variable`, `delphin.predicate`, `delphin.lnk`, etc.) was partly to make a place for relevant documentation
- Docs are versioned with PyDelphin releases (v0.9.2, v1.0.0) and to the latest commit
- https://pydelphin.rtfd.io/

1.3 Processor Interfaces

[30]:
```
from delphin import ace                    # formerly delphin.interfaces.ace

grm = '~/grammars/erg-2018-x86-64-0.9.30.dat'

ace.parse(grm, 'Abrams cheered.')['readings']
```

[30]:
```
4
```

[31]:
```
from delphin.web import client             # formerly delphin.interfaces.rest
response = client.parse('Browne gasped!')
result = response.result(0)
```

[32]:
```
result['eds']
```

[32]:
```
{'top': 'e3',
 'nodes': {'_1': {'label': 'proper_q',
   'lnk': {'from': 0, 'to': 6},
   'edges': {'BV': 'x6'}},
  'x6': {'label': 'named',
   'lnk': {'from': 0, 'to': 6},
   'carg': 'Browne',
   'type': 'x',
   'properties': {'PERS': '3', 'NUM': 'sg', 'IND': '+'},
   'edges': {}},
  'e3': {'label': '_gasp_v_1',
   'lnk': {'from': 7, 'to': 14},
   'type': 'e',
   'properties': {'SF': 'prop',
    'TENSE': 'past',
    'MOOD': 'indicative',
    'PROG': '-',
    'PERF': '-'},
   'edges': {'ARG1': 'x6'}}}}
```

[33]:
```
result.eds()
```

[33]:
```
<EDS object (proper_q named _gasp_v_1) at 140681210863376>
```

1.4 Web Server

- Subset of ErgAPI (only JSON formats, fewer options)
- Also a superset of the ErgAPI (+generation, +test suite browsing)
- Very easy configuration

```
import falcon
from delphin.web import server

application = falcon.API()
application.req_options.strip_url_path_trailing_slash = True

server.configure(
    application,
    parser='~/grammars/erg-2018-x86-64-0.9.30.dat',
    generator='~/grammars/erg-2018-x86-64-0.9.30.dat',
    testsuites={
        "data": [
            {'name': 'mrs', 'path': '~/grammars/erg-trunk/tsdb/gold/mrs'}
        ]
    }
)
```

[34]:
```
from delphin.web import client
server = 'http://127.0.0.1:8000/'
response = client.parse('Kim punted.', params={'mrs': True}, server=server)
response.result(0).mrs()
```

[34]:
```
<MRS object (proper_q named _punted/vbd_u_unknown) at 140681210975720>
```

1.5 Semantics

- MRS, DMRS, and EDS are *independent* (no more `Xmrs`)

[35]:
```
from delphin import dmrs  # formerly delphin.mrs.dmrs
m = ace.parse(grm, 'Kim thought Sandy had arrived and left.').result(0).mrs()
d = dmrs.from_mrs(m)  # conversion step is necessary (no shared representation)
d.nodes
```

[35]:
```
[<Node object (10000:proper_q<0:3>) at 140681211230624>,
 <Node object (10001:named<0:3>) at 140681210618760>,
 <Node object (10002:_think_v_1<4:11>) at 140681210618344>,
 <Node object (10003:proper_q<12:17>) at 140681210618240>,
 <Node object (10004:named<12:17>) at 140681210618136>,
 <Node object (10005:_arrive_v_1<22:29>) at 140681210618448>,
 <Node object (10006:_and_c<30:33>) at 140681210618032>,
 <Node object (10007:_leave_v_1<34:39>) at 140681210619592>]
```

[36]:
```
d.links
```

[36]:
```
[<Link object (10000 :RSTR/H 10001) at 140681211633952>,
 <Link object (10002 :ARG1/NEQ 10001) at 140681210605928>,
 <Link object (10002 :ARG2/H 10005) at 140681210606432>,
 <Link object (10003 :RSTR/H 10004) at 140681210545064>,
 <Link object (10005 :ARG1/NEQ 10004) at 140681210546000>,
 <Link object (10006 :ARG1/EQ 10005) at 140681210546072>,
 <Link object (10006 :ARG2/EQ 10007) at 140681210546144>,
 <Link object (10007 :ARG1/NEQ 10004) at 140681210546216>,
 <Link object (10007 :MOD/EQ 10005) at 140681210546288>]
```

[37]:
```
d.arguments()
```

[37]:
```
{10000: {'RSTR': 10001},
 10001: {},
 10002: {'ARG1': 10001, 'ARG2': 10005},
 10003: {'RSTR': 10004},
 10004: {},
 10005: {'ARG1': 10004},
 10006: {'ARG1': 10005, 'ARG2': 10007},
 10007: {'ARG1': 10004}}
```

[38]:
```
d.arguments(scopal=True)
```

[38]:
```
{10000: {'RSTR': 10001},
 10001: {},
 10002: {'ARG2': 10005},
 10003: {'RSTR': 10004},
 10004: {},
 10005: {},
 10006: {},
 10007: {}}
```

1.6 Scope Operations

[39]:
```
d.scopes()
```

[39]:
```
{'h0': [],
 'h1': [10000],
 'h2': [10001],
 'h3': [10002],
 'h4': [10003],
 'h5': [10004],
 'h6': [10005, 10007, 10006]}
```

[40]:
```
d.scope_constraints()
```

[40]:
```
[('h0', 'qeq', 'h3'),
 ('h1', 'qeq', 'h2'),
 ('h3', 'qeq', 'h6'),
 ('h4', 'qeq', 'h5')]
```

[41]:
```
from delphin import scope
scope.representatives(d)
```

[41]:
```
{10002: [],
 'h0': [],
 'h1': [10000],
 'h2': [10001],
 'h3': [10002],
 'h4': [10003],
 'h5': [10004],
 'h6': [10005, 10007]}
```

[42]:
```
scope.tree_fragments(d)
```

[42]:
```
{10002: UnderspecifiedScope(set(), {}, {}),
 'h0': UnderspecifiedScope(set(), {}, {'h3': UnderspecifiedScope({10002}, {},
{'h6': UnderspecifiedScope({10005, 10006, 10007}, {}, {})})}),
 'h1': UnderspecifiedScope({10000}, {}, {'h2': UnderspecifiedScope({10001}, {},
{})}),
 'h4': UnderspecifiedScope({10003}, {}, {'h5': UnderspecifiedScope({10004}, {},
{})})}
```

1.7 Test Suites

[43]:
```
from delphin import itsdb
ts = itsdb.TestSuite('~/grammars/jacy/tsdb/gold/mrs')
ts.schema['item']
```

[43]:
```
[<delphin.tsdb.Field at 0x7ff2e5999f68>,
 <delphin.tsdb.Field at 0x7ff2e5999eb8>,
 <delphin.tsdb.Field at 0x7ff2e5999e08>,
 <delphin.tsdb.Field at 0x7ff2e5999d00>,
 <delphin.tsdb.Field at 0x7ff2e5999d58>,
 <delphin.tsdb.Field at 0x7ff2e593faf0>,
 <delphin.tsdb.Field at 0x7ff2e593fb48>,
 <delphin.tsdb.Field at 0x7ff2e593fba0>,
 <delphin.tsdb.Field at 0x7ff2e593fbf8>,
 <delphin.tsdb.Field at 0x7ff2e593fc50>,
 <delphin.tsdb.Field at 0x7ff2e593fca8>,
 <delphin.tsdb.Field at 0x7ff2e593fd00>,
 <delphin.tsdb.Field at 0x7ff2e593fd58>,
 <delphin.tsdb.Field at 0x7ff2e593fdb0>,
 <delphin.tsdb.Field at 0x7ff2e593fe08>]
```

1.7.1 Performance optimizations to help with Rademaker-profiles

- low memory usage
- delayed parsing

[44]:
```
ts['item'][-1]
```

[44]:
```
<Row object (1071, 'unknown', 'formal', 'none', 1, 'S', '      ',
None, None, None, 1, 7, 'The dog arrived barking.', 'bond',
datetime.datetime(2006, 5, 28, 0, 0)) at 140681210816712>
```

1.7.2 "Transactional"

[45]:
```
ts.in_transaction
```

[45]:
```
False
```

[46]:
```
ts['item'].update(0, {'i-input': '    '})
ts.in_transaction
```

[46]:
```
True
```

[47]:
```
ts.commit()
ts.in_transaction
```

[47]:
```
False
```

1.7.3 A more user-friendly view

[48]:
```
r = next(ts.processed_items())
len(r.results())
```

[48]:
```
1
```

[49]:
```
print(r.result(0).derivation().to_udf())
```

```
(utterance-root
 (91 utterance_rule-decl-finite -0.723739 0 4
  (90 head_subj_rule -1.05796 0 4
   (87 hf-complement-rule -0.50201 0 2
    (86 quantify-n-rule -0.32216 0 1
     (5 ame-noun 0 0 1
      (""
       1 "\"\"")))
    (6 ga 0.531537 1 2
     (""
      2 "\"\"")))
   (89 vstem-vend-rule -0.471785 2 4
    (88 t-lexeme-c-stem-infl-rule 0.120963 2 3
     (14 furu_1 0 2 3
      (""
       3 "\"\"")))
    (24 ta-end -0.380719 3 4
     (""
      4 "\"\"")))))))
```