

PLOVER:

A new framework for political event data

Philip A. Schrodt

Parus Analytics LLC and Open Event Data Alliance

Charlottesville, VA USA

<http://philipschrodt.org>

<https://github.com/openeventdata/PLOVER>

Paper presented at the
European Political Science Association meetings, Milan
22 June 2017

PARUS

ANALYTICS



Event Data: Core Innovation

Once calibrated, monitoring and forecasting models based on real-time event data can be run entirely without human intervention

- ▶ Web-based news feeds provide a rich multi-source flow of political information in real time
- ▶ Statistical models can be run and tested automatically, and are 100% transparent

In other words, for the first time in human history—quite literally—we have a system that can provide real-time measures of political activity without any human intermediaries

Major phases of event data

- ▶ 1960s-70s: Original development by Charles McClelland (WEIS; DARPA funding) and Edward Azar (COPDAB; CIA funding?). Focus, then as now, is crisis forecasting.
- ▶ 1980s: Various human coding efforts, including Richard Beale in National Security Council, unsuccessfully attempt to get near-real-time coverage from major newspapers
- ▶ 1990s: KEDS (Kansas) automated coder; PANDA project (Harvard) extends ontologies to sub-state actions; shift to wire service data
- ▶ early 2000s: TABARI and VRA second-generation automated coders
- ▶ 2007-2011: DARPA ICEWS
- ▶ 2012-present: full-parsing coders from near-real-time web-based news sources: PETRARCH and ACCENT

Development of event ontologies

1970s: WEIS, COPDAB, CREON and others

1980s: BCOW (Leng) (crisis data: 300 categories)

1990s: PANDA (Bond): first ontology to focus on substate actors

2000s: IDEA (Bond, VRA): backward compatible with multiple existing ontologies, adds non-political events such as disaster and disease

2000s: CAMEO (Gerner and Schrodtr): combines ambiguous WEIS categories, expands violence and mediation-related categories; implemented as 15,000-phrase TABARI dictionary

late 2010s: PLOVER: generalized political coding scheme and data interchange specification

WEIS primary categories (ca. 1965)

01	Yield	11	Reject
02	Comment	12	Accuse
03	Consult	13	Protest
04	Approve	14	Deny
05	Promise	15	Demand
06	Grant	16	Warn
07	Reward	17	Threaten
08	Agree	18	Demonstrate
09	Request	19	Reduce Relationship
10	Propose	20	Expel
		21	Seize
		22	Force

CAMEO

- ▶ 20 primary event categories; around 200 subcategories
- ▶ Based on the WEIS typology but with greater detail on violence and mediation
- ▶ Combines ambiguous WEIS categories such as [WARN/THREATEN] and [GRANT/PROMISE]
- ▶ National actor codes based on ISO-3166 and `CountryInfo.txt`
- ▶ Substate “agents” such as GOV, MIL, REB, BUS
- ▶ Extensive IGO/NGO list

Open Event Data Alliance

- ▶ Institutionalize event data following the model of CRAN and many other decentralized open collaborative research groups: these turn out to be common in most research communities
- ▶ Provide at least one source of daily updates with 24/7/365 data reliability. Ideally, multiple such data sets rather than “one data set to rule them all”
- ▶ Establish common standards, formats, and best practices
- ▶ Open source, open collaboration, open access

PLOVER

Political Language Ontology for Verifiable Event Records
Event, Actor and Data Interchange Specification

Open Event Data Alliance

<http://openeventdata.org/>

<http://ploverdata.org/>

DRAFT Version: 0.6b2

March 2017



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

PLOVER objectives

- ▶ Only the 2-digit event “cue categories” have been retained from CAMEO. These are defined in greater detail than they were in WEIS and CAMEO.
- ▶ Some additional consolidation of CAMEO codes, and a new category for criminal behavior
- ▶ Standard optional fields have been defined for some categories, and the “target” is optional in some categories.
- ▶ A set of standardized names (“fields”) for JSON (<http://www.json.org/>) records are specified for both the core event data fields and for extended information such as geolocation and extracted texts;
- ▶ We have converted all of the examples in the CAMEO manual to an initial set of English-language “gold standard records” for validation purposes—these files are at https://github.com/openeventdata/PLOVER/blob/master/PLOVER_GSR_CAMEO.txt—and we expect to both expand this corpus and extend it to at least Spanish and Arabic cases.

Event, Mode, and Context

Most of the detail found in the 3- and 4-digit categories of CAMEO is now found in the *mode* and *context* fields in PLOVER. More generally, PLOVER takes the general purpose “events” of CAMEO (as well as the earlier WEIS, IDEA and COPDAB ontologies) and splits these into “*event – mode – context*” which generally corresponds to “*what – how – why*.” We anticipate at least four advantages to this:

1. The “*what – how – why*” components are now distinct, whereas various CAMEO subcategories inconsistently used the *how* and *why* to distinguish between subcategories.
2. We are probably increasing the ability of automated classifiers—as distinct from parser/coders—to assign *mode* and *context* compared to their ability to assign subcategories.
3. In initial experiments, it appears this approach is *much* easier for humans to code than the hierarchical structure of CAMEO because a human coder can hold most of the relevant categories in working memory (well, that and a few tables easily displayed on a screen)
4. Because the words used in differentiate *mode* and *context* are generally very basic, translations of the coding protocols into languages other than English is likely to be easier than translating the subcategory descriptions found in CAMEO.

PLOVER: ASSAULT modes

Name	Content
beat	physically assault
torture	torture
execute	judicially-sanctioned execution
sexual	sexual violence
assassinate	targeted assassinations with any weapon
primitive	primitive weapons: fire, edged weapons, rocks, farm implements
firearms	rifles, pistols, light machine guns
explosives	any explosive not incorporated in a heavy weapon: mines, IEDS, car b
suicide-attack	individual and vehicular suicide attacks
heavy-weapons	crew-served weapons
other	other modes

Adapted from Political Instability Task Force Atrocities Database:

<http://eventdata.parusanalytics.com/data.dir/atrocities.html>

PLOVER: general contexts

Name	Content
political	political contexts not covered by any of the more specific categories below
military	military, including military assistance
economic	trade, finance and economic development
diplomatic	diplomacy
resource	territory and natural resources
culture	cultural and educational exchange
disease	disease outbreaks and epidemics
disaster	natural disaster
refugee	refugees and forced migration
legal	national and international law, including human rights
terrorism	terrorism
government	governmental issues other than elections and legislative
election	elections and campaigns
legislative	legislative debate, parliamentary coalition formation
cbrn	chemical, biological, radiation, and nuclear attacks
cyber	cyber attacks and crime
historical	event is historical
hypothetical	event is hypothetical

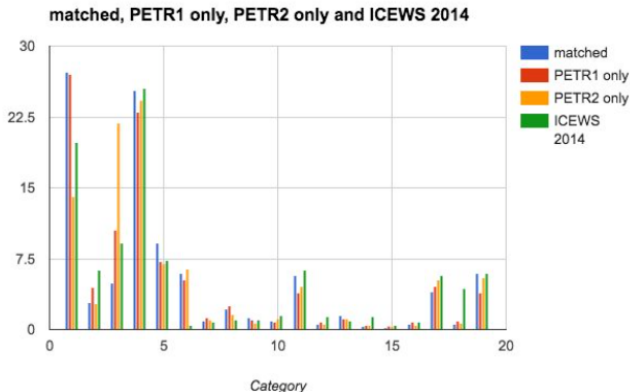
PLOVER output

```
{  
  "id": "test-0056-0036_1",  
  "date": "2015-02-12",  
  "source": [{ "actorText": "Russian Foreign Minister Sergei Lavrov", "code": "RUS", "sector": "GOV"},  
             { "actorText": "Iranian counterpart Mohammad Javad Zarif", "code": "IRN"}],  
  "target": [{ "actorText": "Syria crisis", "code": "SYR"}],  
  "event": "DISCUSS",  
  "eventText": "discussed",  
  "mode": "mode-holder",  
  "context": "context-holder",  
  "text": "MOSCOW: Russian Foreign Minister Sergei Lavrov and his Iranian counterpart Mohammad Javad  
  Zarif discussed the Syria crisis by phone Wednesday, the Russian Foreign Ministry said in a statement",  
  "language": "en",  
  "publication": "mudflat test data",  
  "coder": "Parus Analytics",  
  "version": "0.5b1",  
  "dateCoded": "2017-03-20",  
  "comment": "test output from mudflat",  
},
```

Event data coding programs

- ▶ TABARI: C/C++ using internal shallow parsing.
<http://eventdata.parusanalytics.com/software.dir/tabari.html>
- ▶ JABARI: Java version of TABARI with additional enhancements: alas, abandoned and lost following end of ICEWS research phase
- ▶ DARPA ICEWS: Raytheon/BBN ACCENT coder can now be licensed for academic research use
- ▶ Open Event Data Alliance: PETRARCH 1/2 coders, Moredcai geolocation system.
<https://github.com/openeventdata>
- ▶ NSF RIDIR: developing open-source native-language coders and dictionaries for English, Spanish and Arabic

“CAMEO-World” across coders and news sources



Between-category variance is massively greater than the between-coder variance.

Why the convergence?

- ▶ This is simply how news is covered (human-coded WEIS data also looked similar)
- ▶ The diversity in the language and formatting of stories means no automated coding system can get all of them
- ▶ Major differences (PETRARCH-2 on 03; ACCENT on 06, 18) are due to redefinitions or intense dictionary development
- ▶ Systems probably have comparable performance on avoiding non-events (95% agreement for PETRARCH 1 and 2)
- ▶ Note these are aggregate *proportions*: ACCENT probably has a higher recall rate, but the otherwise pattern is still the same

So...

philip-schrodt / mudflat

Unwatch

1

Star

0

Fork

0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Pulse

Graphs

Settings

No description, website, or topics provided.

Edit

Add topics

25 commits

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

philip-schrodt committed on GitHub Added doc for CAMEO2PLOVER.txt Latest commit f6a489a an hour ago

CAMEO2PLOVER.txt	PLOVER event-mode-context equivalents for CAMEO codes	an hour ago
README.md	Added doc for CAMEO2PLOVER.txt	an hour ago
coder.py	Simplified/obscured get_NP() and get_conj() with list comprehensions	18 hours ago
extract_UD_parse.py	Add primitive version of get_nsubj()	13 days ago
globals.py	Add coder module; rename reader and globals modules	11 days ago
mudflat.py	Coding for basic compounds and agents	11 days ago
mudflat_testdata_Mk1.txt	Coding for basic compounds and agents	11 days ago
reader.py	Coding for basic compounds and agents	11 days ago
utilities.py	Coding for basic compounds and agents	11 days ago

README.md

mudflat

Minimal universal dependency friendly little automated tagger

A coding system supporting PLOVER (of course): <https://github.com/openeventdata/PLOVER>; <http://ploverdata.org>

Universal dependencies

Universal Dependencies v2

Executive summary of changes from v1 to v2

- Tokenization and word segmentation
- Morphology
 - General principles
 - Universal POS tags (single document)
 - Universal features (single document)
 - Language-specific features
 - Conversion from other tagsets
- Syntax
 - General principles
 - Basic dependencies
 - Simple clauses
 - Nominals
 - Complex clauses
 - Other constructions
 - Enhanced dependencies
 - Universal dependency relations (single document)
 - Language-specific relations
- CoNLL-U format

This is the online documentation for Universal Dependencies, version 2 (2016-12-01). **Note:** The treebanks listed below still follow the v1 guidelines available [here](#).

Upcoming UD-related events

- [CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies](#)
- [EACL 2017 Tutorial on Universal Dependencies](#)
- [NoDaLiDa Workshop on Universal Dependencies \(UDW 2017\)](#)

Want to know more about UD?

- [Short introduction to Universal Dependencies](#)
- [How to contribute to UD](#)
- [Tools for working with UD](#)

If you want to receive news about Universal Dependencies, you can subscribe to the [UD mailing list](#).

UD Treebanks

[illegible]

Dependency parse: input

```
# sent_id = test-0056-0036_1
# source = mudflat test data
# date = 2015-02-12
# text = MOSCOW: Russian Foreign Minister Sergei Lavrov and his Iranian counterpart
# text = counterpart Mohammad Javad Zarif discussed the Syria crisis by phone
# text = Wednesday, the Russian Foreign Ministry said in a statement.
1  MOSCOW  MOSCOW  _  NNP  _  0  root  _  _
2  :  :  _  _  1  punct  _  _
3  Russian  Russian  _  NNP  _  7  compound  _  _
4  Foreign  Foreign  _  NNP  _  7  compound  _  _
5  Minister  Minister  _  NNP  _  7  compound  _  _
6  Sergei  Sergei  _  NNP  _  7  compound  _  _
7  Lavrov  Lavrov  _  NNP  _  15  nsubj  _  _
8  and  and  _  CC  _  7  cc  _  _
9  his  he  _  PRP$  _  14  nmod:poss  _  _
10 Iranian  iranian  _  JJ  _  14  amod  _  _
11 counterpart  counterpart  _  NN  _  14  compound  _  _
12 Mohammad  Mohammad  _  NNP  _  14  compound  _  _
13 Javad  Javad  _  NNP  _  14  compound  _  _
14 Zarif  Zarif  _  NNP  _  7  conj  _  _
15 discussed  discuss  _  VBD  _  1  dep  _  _
16 the  the  _  DT  _  18  det  _  _
17 Syria  Syria  _  NNP  _  18  compound  _  _
18 crisis  crisis  _  NN  _  15  dobj  _  _
19 by  by  _  IN  _  20  case  _  _
20 phone  phone  _  NN  _  15  nmod  _  _
21 Wednesday  Wednesday  _  NNP  _  15  nmod:tmod  _  _
22 ,  ,  _  _  15  punct  _  _
23 the  the  _  DT  _  26  det  _  _
24 Russian  Russian  _  NNP  _  26  compound  _  _
25 Foreign  Foreign  _  NNP  _  26  compound  _  _
26 Ministry  Ministry  _  NNP  _  27  nsubj  _  _
27 said  say  _  VBD  _  15  parataxis  _  _
28 in  in  _  IN  _  30  case  _  _
29 a  a  _  DT  _  30  det  _  _
30 statement  statement  _  NN  _  27  nmod  _  _
31 .  .  _  _  1  punct  _  _
```

Dependency parse: locate subject

```
# sent_id = test-0056-0036_1
# source = mudflat test data
# date = 2015-02-12
# text = MOSCOW: Russian Foreign Minister Sergei Lavrov and his Iranian counterpart
# text = counterpart Mohammad Javad Zarif discussed the Syria crisis by phone
# text = Wednesday, the Russian Foreign Ministry said in a statement.
1 MOSCOW MOSCOW _ NNP _ 0 root _ _
2 : : _ : _ 1 punct _ _
3 Russian Russian _ NNP _ 7 compound _ _
4 Foreign Foreign _ NNP _ 7 compound _ _
5 Minister Minister _ NNP _ 7 compound _ _
6 Sergei Sergei _ NNP _ 7 compound _ _
7 Lavrov Lavrov _ NNP _ 15 nsubj _ _
8 and and _ CC _ 7 cc _ _
9 his he _ PRP$ _ 14 nmod:poss _ _
10 Iranian iranian _ JJ _ 14 amod _ _
11 counterpart counterpart _ NN _ 14 compound _ _
12 Mohammad Mohammad _ NNP _ 14 compound _ _
13 Javad Javad _ NNP _ 14 compound _ _
14 Zarif Zarif _ NNP _ 7 conj _ _
15 discussed discuss _ VBD _ 1 dep _ _
16 the the _ DT _ 18 det _ _
17 Syria Syria _ NNP _ 18 compound _ _
18 crisis crisis _ NN _ 15 dobj _ _
19 by by _ IN _ 20 case _ _
20 phone phone _ NN _ 15 nmod _ _
21 Wednesday Wednesday _ NNP _ 15 nmod:tmod _ _
22 , , _ , _ 15 punct _ _
23 the the _ DT _ 26 det _ _
24 Russian Russian _ NNP _ 26 compound _ _
25 Foreign Foreign _ NNP _ 26 compound _ _
26 Ministry Ministry _ NNP _ 27 nsubj _ _
27 said say _ VBD _ 15 parataxis _ _
28 in in _ IN _ 30 case _ _
29 a a _ DT _ 30 det _ _
30 statement statement _ NN _ 27 nmod _ _
31 . . _ . _ 1 punct _ _
```

Dependency parse: locate verb

```
# sent_id = test-0056-0036_1
# source = mudflat test data
# date = 2015-02-12
# text = MOSCOW: Russian Foreign Minister Sergei Lavrov and his Iranian counterpart
# text = counterpart Mohammad Javad Zarif discussed the Syria crisis by phone
# text = Wednesday, the Russian Foreign Ministry said in a statement.
1 MOSCOW MOSCOW _ NNP _ 0 root _ _
2 : : _ : _ 1 punct _ _
3 Russian Russian _ NNP _ 7 compound _ _
4 Foreign Foreign _ NNP _ 7 compound _ _
5 Minister Minister _ NNP _ 7 compound _ _
6 Sergei Sergei _ NNP _ 7 compound _ _
7 Lavrov Lavrov _ NNP _ 15 nsubj _ _
8 and and _ CC _ 7 cc _ _
9 his he _ PRP$ _ 14 nmod:poss _ _
10 Iranian iranian _ JJ _ 14 amod _ _
11 counterpart counterpart _ NN _ 14 compound _ _
12 Mohammad Mohammad _ NNP _ 14 compound _ _
13 Javad Javad _ NNP _ 14 compound _ _
14 Zarif Zarif _ NNP _ 7 conj _ _
15 discussed discuss _ VBD _ 1 dep _ _
16 the the _ DT _ 18 det _ _
17 Syria Syria _ NNP _ 18 compound _ _
18 crisis crisis _ NN _ 15 dobj _ _
19 by by _ IN _ 20 case _ _
20 phone phone _ NN _ 15 nmod _ _
21 Wednesday Wednesday _ NNP _ 15 nmod:tmod _ _
22 , , _ , _ 15 punct _ _
23 the the _ DT _ 26 det _ _
24 Russian Russian _ NNP _ 26 compound _ _
25 Foreign Foreign _ NNP _ 26 compound _ _
26 Ministry Ministry _ NNP _ 27 nsubj _ _
27 said say _ VBD _ 15 parataxis _ _
28 in in _ IN _ 30 case _ _
29 a a _ DT _ 30 det _ _
30 statement statement _ NN _ 27 nmod _ _
31 . . _ . _ 1 punct _ _
```

Dependency parse: locate direct object

```
# sent_id = test-0056-0036_1
# source = mudflat test data
# date = 2015-02-12
# text = MOSCOW: Russian Foreign Minister Sergei Lavrov and his Iranian counterpart
# text = counterpart Mohammad Javad Zarif discussed the Syria crisis by phone
# text = Wednesday, the Russian Foreign Ministry said in a statement.
1 MOSCOW MOSCOW _ NNP _ 0 root _ _
2 : : _ : _ 1 punct _ _
3 Russian Russian _ NNP _ 7 compound _ _
4 Foreign Foreign _ NNP _ 7 compound _ _
5 Minister Minister _ NNP _ 7 compound _ _
6 Sergei Sergei _ NNP _ 7 compound _ _
7 Lavrov Lavrov _ NNP _ 15 nsubj _ _
8 and and _ CC _ 7 cc _ _
9 his he _ PRP$ _ 14 nmod:poss _ _
10 Iranian iranian _ JJ _ 14 amod _ _
11 counterpart counterpart _ NN _ 14 compound _ _
12 Mohammad Mohammad _ NNP _ 14 compound _ _
13 Javad Javad _ NNP _ 14 compound _ _
14 Zarif Zarif _ NNP _ 7 conj _ _
15 discussed discuss _ VBD _ 1 dep _ _
16 the the _ DT _ 18 det _ _
17 Syria Syria _ NNP _ 18 compound _ _
18 crisis crisis _ NN _ 15 dobj _ _
19 by by _ IN _ 20 case _ _
20 phone phone _ NN _ 15 nmod _ _
21 Wednesday Wednesday _ NNP _ 15 nmod:tmod _ _
22 , , _ , _ 15 punct _ _
23 the the _ DT _ 26 det _ _
24 Russian Russian _ NNP _ 26 compound _ _
25 Foreign Foreign _ NNP _ 26 compound _ _
26 Ministry Ministry _ NNP _ 27 nsubj _ _
27 said say _ VBD _ 15 parataxis _ _
28 in in _ IN _ 30 case _ _
29 a a _ DT _ 30 det _ _
30 statement statement _ NN _ 27 nmod _ _
31 . . _ . _ 1 punct _ _
```

Dependency parse: locate actor phrases

```
# sent_id = test-0056-0036_1
# source = mudflat test data
# date = 2015-02-12
# text = MOSCOW: Russian Foreign Minister Sergei Lavrov and his Iranian counterpart
# text = counterpart Mohammad Javad Zarif discussed the Syria crisis by phone
# text = Wednesday, the Russian Foreign Ministry said in a statement.
1 MOSCOW MOSCOW _ NNP _ 0 root _ _
2 : : _ : _ 1 punct _ _
3 Russian Russian _ NNP _ 7 compound _ _
4 Foreign Foreign _ NNP _ 7 compound _ _
5 Minister Minister _ NNP _ 7 compound _ _
6 Sergei Sergei _ NNP _ 7 compound _ _
7 Lavrov Lavrov _ NNP _ 15 nsubj _ _
8 and and _ CC _ 7 cc _ _
9 his he _ PRP$ _ 14 nmod:poss _ _
10 Iranian iranian _ JJ _ 14 amod _ _
11 counterpart counterpart _ NN _ 14 compound _ _
12 Mohammad Mohammad _ NNP _ 14 compound _ _
13 Javad Javad _ NNP _ 14 compound _ _
14 Zarif Zarif _ NNP _ 7 conj _ _
15 discussed discuss _ VBD _ 1 dep _ _
16 the the _ DT _ 18 det _ _
17 Syria Syria _ NNP _ 18 compound _ _
18 crisis crisis _ NN _ 15 dobj _ _
19 by by _ IN _ 20 case _ _
20 phone phone _ NN _ 15 nmod _ _
21 Wednesday Wednesday _ NNP _ 15 nmod:tmod _ _
22 , , _ , _ 15 punct _ _
23 the the _ DT _ 26 det _ _
24 Russian Russian _ NNP _ 26 compound _ _
25 Foreign Foreign _ NNP _ 26 compound _ _
26 Ministry Ministry _ NNP _ 27 nsubj _ _
27 said say _ VBD _ 15 parataxis _ _
28 in in _ IN _ 30 case _ _
29 a a _ DT _ 30 det _ _
30 statement statement _ NN _ 27 nmod _ _
31 . . _ . _ 1 punct _ _
```

The image displays a dependency parse tree for the sentence: "discussed the Syria crisis by phone Wednesday, the Russian Foreign Ministry said in a statement." The tree is structured as a list of tokens with their grammatical roles and dependencies. Two dependencies are highlighted with arrows and circles:

- A blue arrow points from the word "discussed" (line 15) to the word "Sergei" (line 6), which is part of the noun phrase "Sergei Lavrov" (lines 6-7). This dependency is labeled "nsubj" (nominal subject).
- A green arrow points from the word "discussed" (line 15) to the word "crisis" (line 18). This dependency is labeled "dobj" (direct object).

Dependency parse: locate phrases linked by conjunction

```
# sent_id = test-0056-0036_1
# source = mudflat test data
# date = 2015-02-12
# text = MOSCOW: Russian Foreign Minister Sergei Lavrov and his Iranian counterpart
# text = counterpart Mohammad Javad Zarif discussed the Syria crisis by phone
# text = Wednesday, the Russian Foreign Ministry said in a statement.
1 MOSCOW MOSCOW NNP 0 root - -
2 : : - 1 punct - -
3 Russian Russian NNP 7 compound - -
4 Foreign Foreign NNP 7 compound - -
5 Minister Minister NNP 7 compound - -
6 Sergei Sergei NNP 7 compound - -
7 Lavrov Lavrov NNP 15 nsubj - -
8 and and CC 7 cc - -
9 his he PRP$ 14 nmod:poss - -
10 Iranian Iranian JJ 14 amod - -
11 counterpart counterpart NN 14 compound - -
12 Mohammad Mohammad NNP 14 compound - -
13 Javad Javad NNP 14 compound - -
14 Zarif Zarif NNP 7 conj - -
15 discussed discuss VBD 1 dep - -
16 the the DT 18 det - -
17 Syria Syria NNP 18 compound - -
18 crisis crisis NN 15 dobj - -
19 by by IN 20 case - -
20 phone phone NN 15 nmod - -
21 Wednesday Wednesday NNP 15 nmod:tmod - -
22 , , - 15 punct - -
23 the the DT 26 det - -
24 Russian Russian NNP 26 compound - -
25 Foreign Foreign NNP 26 compound - -
26 Ministry Ministry NNP 27 nsubj - -
27 said say VBD 15 parataxis - -
28 in in IN 30 case - -
29 a a DT 30 det - -
30 statement statement NN 27 nmod - -
31 . . - 1 punct - -
```

The image displays a dependency parse tree for the sentence: "MOSCOW: Russian Foreign Minister Sergei Lavrov and his Iranian counterpart Mohammad Javad Zarif discussed the Syria crisis by phone Wednesday, the Russian Foreign Ministry said in a statement." The tree is represented as a list of tokens with their grammatical roles and dependency relations. Key annotations include:

- A red circle around the node for "Lavrov" (line 7) with the role "nsubj".
- A green circle around the node for "crisis" (line 18) with the role "dobj".
- A blue circle around the node for "discussed" (line 15) with the role "dep".
- A yellow circle around the node for "Zarif" (line 14) with the role "conj".
- A yellow arrow pointing from the "discussed" node (line 15) to the "Zarif" node (line 14).
- A blue arrow pointing from the "discussed" node (line 15) to the "crisis" node (line 18).

Main event coding: mudflat

```
def get_NP(sdex):
    """ construct noun phrase based on word at sdex """
    index = int(sdex) - 1
    subjstrg = plist[index][1]
    for li in reversed(plist[:index]):
        if li[6] == sdex and li[7] in ["compound", "amod"]:
            subjstrg = li[1] + ' ' + subjstrg
    for li in plist[index + 1:]: # do we ever hit this?
        if li[6] == sdex and li[7] in ["compound", "amod"]:
            subjstrg = subjstrg + ' ' + li[1]
    return subjstrg

def get_conj(sdex):
    """ check if there are compound elements: this can be reduced to a, well, reduce """
    actlist = [sdex]
    for li in plist:
        if li[6] == sdex and li[7] == "conj":
            actlist.append(li[0])
    return actlist

def code_events():
    # <same initialization code>
    for li in plist:
        if "nsubj" == li[7]:
            srclist = get_conj(li[0])
            iroot = int(li[6])
            rootcode = plist[iroot - 1][2].upper() # adjust for zero indexing
            roottext = plist[iroot - 1][1]
            tarlist = []
            for lobj in plist:
                if lobj[7] == "dobj" and lobj[6] == li[6]:
                    tarlist = get_conj(lobj[0])
                if tarlist: break
```

Main event coding: mudflat

```
def get_NP(sdex):
    """ construct noun phrase based on word at sdex """
    index = int(sdex) - 1
    return ' '.join(reversed(
        [li[1] for li in reversed(plist[:index]) if li[6] == sdex and li[7] in ["compound", "amod"]]
    )) + ' ' + plist[index][1] + ' ' + \
        ' '.join([li[1] for li in plist[index + 1:] if li[6] == sdex and li[7] in ["compound", "amod"]])

def get_conj(sdex):
    """ check if there are compound elements """
    return [sdex] + [li[0] for li in plist if li[6] == sdex and li[7] == "conj"]

def code_events():
    """ main coding loop """
    srctext, srccode, srcseccode, srclist = [], [], [], []
    tartext, tarcode, tarseccode, tarlist = [], [], [], []
    roottext, rootcode = "", ""

    for li in plist:
        if "nsubj" == li[7]:
            srclist = get_conj(li[0])
            iroot = int(li[6])
            rootcode = plist[iroot - 1][2].upper() # adjust for zero indexing
            roottext = plist[iroot - 1][1]
            tarlist = []
            for lobj in plist:
                if lobj[7] == "dobj" and lobj[6] == li[6]:
                    tarlist = get_conj(lobj[0])
            if tarlist: break
```

Thank you

Email:

`schrodt735@gmail.com`

Slides:

`http://eventdata.parusanalytics.com/presentations.html`

Links to data and software:

`https://github.com/openeventdata/PLOVER`