A Practical Guide to Current Developments in Event Data

Philip A. Schrodt

Parus Analytics LLC and Open Event Data Alliance
Charlottesville, VA
http://philipschrodt.org
http://eventdata.parusanalytics.com

International Methodology Colloquium
Rice University Webinar
http://www.methods-colloquium.com/
31 March 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

Major technological changes

- late 1980s: Availability of machine-readable news articles
- 2000s: Open source software for natural language processing, machine-learning and time-series statistics
- mid 2000s: Web-based general knowledge resources such as geonames.org and Wikipedia
- late 2000s: Massive expansion of news sources available on the Web
- entire period: Moore's Law

Event data are well suited for predicting political change at short time horizons

- Structural indicators such as GDP, infant mortality, past or adjacent conflict change too slowly
  - They nonetheless affect the overall probability
- Social media indicators change too quickly
  - Social media appear to give—at best—about a six to twenty-four hour warning in collective action situations (Carley; OSI EMBERS)
  - So far, there are no indications that social media provide reliable indicators of deep social/cultural change: signal-to-noise ratio is very low
  - Many authoritarian regimes now extensively manipulate social media with increasingly sophisticated software
- Newsworthy events are "just right"
  - And we've got the models to prove it
  - Which is why they are "newsworthy"
  - Structural indicators either are reflected in the patterns of events, or can be additional covariates

Is automated coding good enough?

- Anyone accustomed to human-coded data and checking records one-by-one will absolutely hate it
- Nonetheless multiple tests have shown it is quite good in statistical and machine learning forecasting applications
  - In particular, it is substitutable for structural indicators
- Cross-project human coding across decades is probably far less accurate than we have led ourselves to believe

For a more extended discussion:
`https://asecondmouse.wordpress.com/2017/02/20/seven-conjectures-on-the-state-of-event-data/`

But fundamentally, comparisons with human coding are irrelevant if one is coding over a billion sentences and updating at the rate of 100,000 stories per day.

News Story Example: Example: 18 December 2007

BAGHDAD. Iraqi leaders criticized Turkey on Monday for bombing Kurdish militants in northern Iraq with airstrikes that they said had left at least one woman dead.

The Turkish attacks in Dohuk Province on Sunday—involving dozens of warplanes and artillery—were the largest known cross-border attack since 2003. They occurred with at least tacit approval from American officials. The Iraqi government, however, said it had not been consulted or informed about the attacks.

Massoud Barzani, leader of the autonomous Kurdish region in the north, condemned the assaults as a violation of Iraqi sovereignty that had undermined months of diplomacy. “These attacks hinder the political efforts exerted to find a peaceful solution based on mutual respect.”

New York Times, 18 December 2007
http://www.nytimes.com/2007/12/18/world/middleeast/18iraq.html?_r=1&ref=world&oref=slogin
(Accessed 18 December 2007)

TABARI Coding: Lead sentence

BAGHDAD. Iraqi leaders criticized Turkey on Monday for bombing Kurdish militants in northern Iraq with airstrikes that they said had left at least one woman dead.

Event Code: 111
Source: IRQ GOV
Target: TUR

Event Code: 223
Source: TUR
Target: IRQKRD REB

TABARI Coding: First event

BAGHDAD. Iraqi leaders criticized Turkey on Monday for bombing Kurdish militants in northern Iraq with airstrikes that they said had left at least one woman dead.

Event Code: 111
Source: IRQ GOV
Target: TUR

Event Code: 223
Source: TUR
Target: IRQKRD REB

TABARI Coding: Actors

BAGHDAD. Iraqi leaders criticized Turkey on Monday for bombing Kurdish militants in northern Iraq with airstrikes that they said had left at least one woman dead.

Event Code: 111
Source: IRQ GOV
Target: TUR

Event Code: 223
Source: TUR
Target: IRQKRD REB

TABARI Coding: Agent

BAGHDAD. Iraqi leaders criticized Turkey on Monday for bombing Kurdish militants in northern Iraq with airstrikes that they said had left at least one woman dead.

Event Code: 111
Source: IRQ GOV
Target: TUR

Event Code: 223
Source: TUR
Target: IRQKRD REB

TABARI Coding: Second event

BAGHDAD. Iraqi leaders criticized Turkey on Monday for bombing Kurdish militants in northern Iraq with airstrikes that they said had left at least one woman dead.

Event Code: 111
Source: IRQ GOV
Target: TUR

Event Code: 223
Source: TUR
Target: IRQKRD REB

TABARI Coding: Second event target

BAGHDAD. Iraqi leaders criticized Turkey on Monday for bombing Kurdish militants in northern Iraq with airstrikes that they said had left at least one woman dead.

Event Code: 111
Source: IRQ GOV
Target: TUR

Event Code: 223
Source: TUR
Target: IRQKRD REB

TABARI Coding: Agent

BAGHDAD. Iraqi leaders criticized Turkey on Monday for bombing Kurdish militants in northern Iraq with airstrikes that they said had left at least one woman dead.

Event Code: 111
Source: IRQ GOV
Target: TUR

Event Code: 223
Source: TUR
Target: IRQKRD REB

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 Schrodt): 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

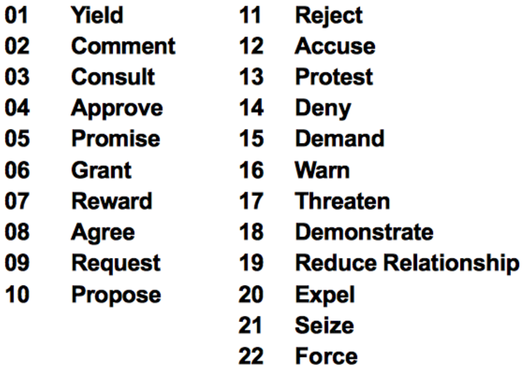
Categorization of Political Interactions

- Distinct English-language verb phrases:
  5,000 to 15,000
  (MUC, KEDS, PANDA projects)

- Micro-level categories
  50 to 200
  (WEIS, BCOW, IDEA, CAMEO)

- Macro-level categories
  10 to 20
  (WEIS, COPDAB, World Handbook, PLOVER)

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

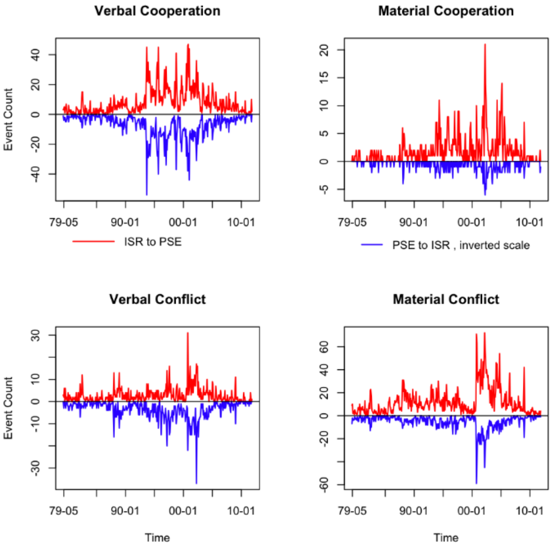
CAMEO

- 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

Quad Counts

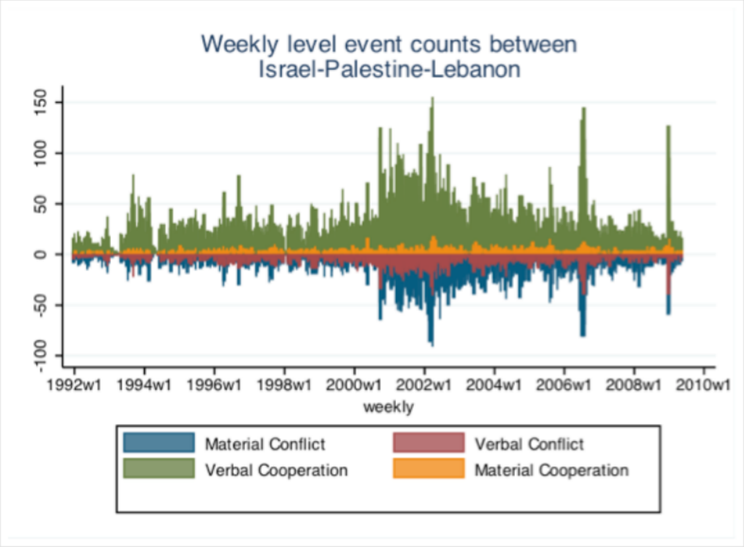
- Verbal Cooperation (VERCP): The occurrence of dialogue-based meetings (i.e. negotiations,peace talks), statements that express a desire to cooperate or appeal for assistance (other than material aid) from other actors. CAMEO categories 01 to 05.
- Material Cooperation (MATCP): Physical acts of collaboration or assistance, including receiving or sending aid, reducing bans and sentencing, etc. CAMEO categories 06 to 09.
- Verbal Conflct (VERCF): A spoken criticism, threat, or accusation, often related to past or future potential acts of material conflct. CAMEO categories 10 to 14.
- Material Conflict (MATCF): Physical acts of a conflictual nature, including armed attacks, destruction of property, assassination, etc. CAMEO categories 15 to 20.

KEDS Project Levant Data, 1979-2010

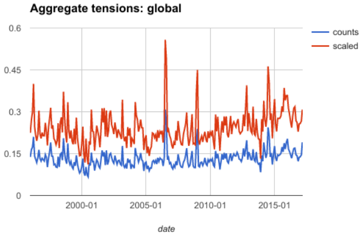


KEDS Project Levant Data, 1992-2010

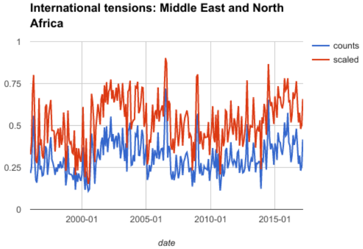
Visualization by Jay Yonamine (Penn State Political Science Ph.D. 2013, now Head of Data Science for Global Patents at Google)



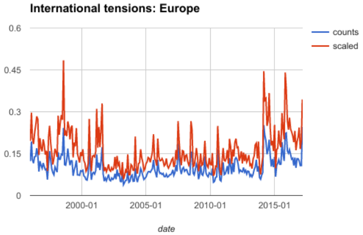
Indicators derived from ICEWS, 1996-2017



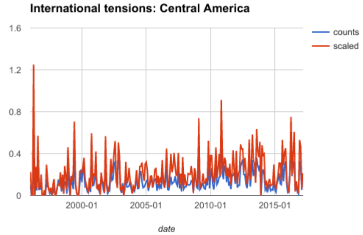
Aggregate tensions: global



International tensions: Middle East and North Africa

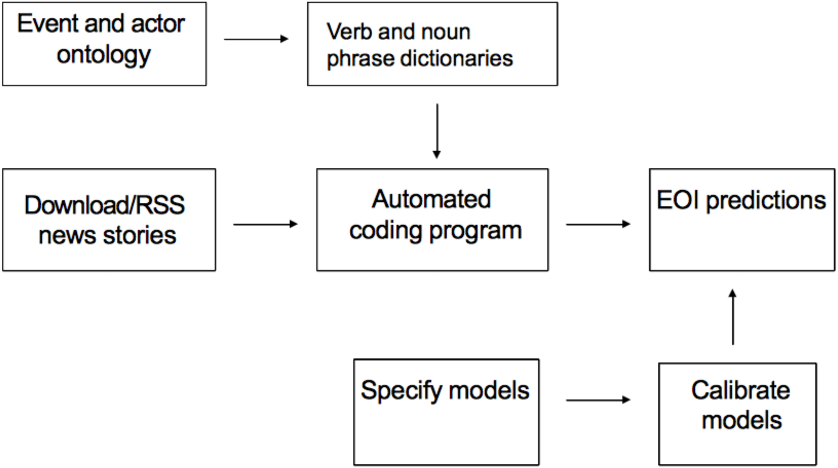


International tensions: Europe



International tensions: Central America

Generating event data: classical approach



Additional steps in contemporary coding

Pre-processing

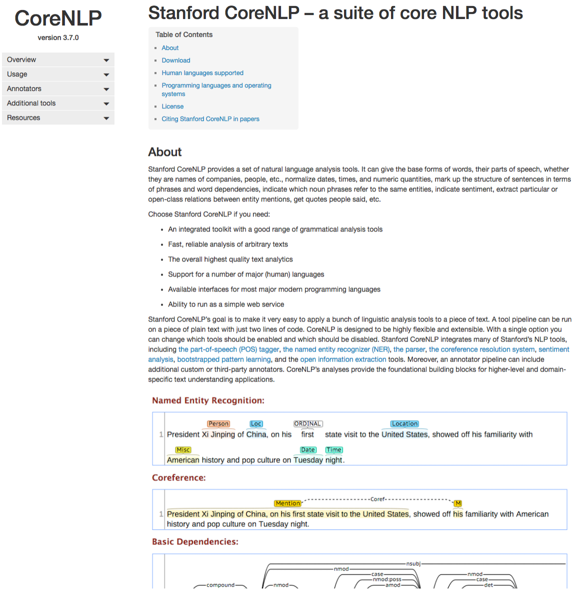
- Filter stories to eliminate sports, movie reviews, business reports etc: a simple SVM is quite effective on this
- Parsing, typically with Stanford CoreNLP
- Clustering similar stories (not in any current pipeline)

Post-processing

- One-a-day filtering, which is a really bad idea except for the alternative of not filtering: see `http://eventdata.parusanalytics.com/papers.dir/Schrodt.TAD-NYU.EventData.pdf`
- Geolocation

These tasks are connected using customized "pipeline" or "glue" programs.

Stanford CoreNLP



CoreNLP

version 3.7.0

- Overview
- Usage
- Annotators
- Additional tools
- Resources

Stanford CoreNLP – a suite of core NLP tools

Table of Contents

About

Stanford CoreNLP provides a set of natural language analysis tools. It can give the base forms of words, their parts of speech, whether they are names of companies, people, etc., normalize dates, times, and numeric quantities, mark up the structure of sentences in terms of phrases and word dependencies, indicate which noun phrases refer to the same entities, indicate sentiment, extract particular or open-class relations between entity mentions, get quotes people said, etc.

Choose Stanford CoreNLP if you need:

- An integrated toolkit with a good range of grammatical analysis tools
- Fast, reliable analysis of arbitrary texts
- The overall highest quality text analytics
- Support for a number of major (human) languages
- Available interfaces for most major modern programming languages
- Ability to run as a simple web service

Stanford CoreNLP's goal is to make it very easy to apply a bunch of linguistic analysis tools to a piece of text. A tool pipeline can be run on a piece of plain text with just two lines of code. CoreNLP is designed to be highly flexible and extensible. With a single option you can change which tools should be enabled and which should be disabled. Stanford CoreNLP integrates many of Stanford's NLP tools, including the part-of-speech (POS) tagger, the named entity recognizer (NER), the parser, the coreference resolution system, sentiment analysis, bootstrapped pattern learning, and the open information extraction tools. Moreover, an annotator pipeline can include additional custom or third-party annotators. CoreNLP's analyses provide the foundational building blocks for higher-level and domain-specific text understanding applications.

Named Entity Recognition:

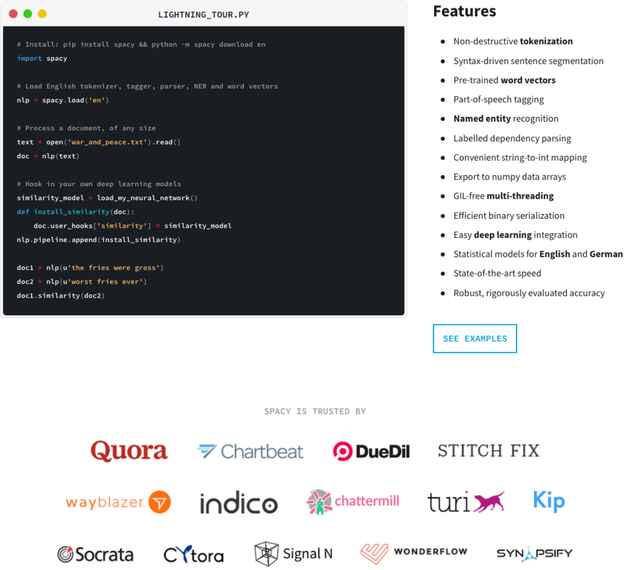
1 President Xi Jinping of China, on his first state visit to the United States, showed off his familiarity with American history and pop culture on Tuesday night.

Coreference:

1 President Xi Jinping of China, on his first state visit to the United States, showed off his familiarity with American history and pop culture on Tuesday night.

Basic Dependencies:

New kid on the block: spaCy



LIGHTNING_TOUR.PY

```
# Install: pip install spacy && python -m spacy download en
import spacy

# Load English tokenizer, tagger, parser, NER and word vectors
nlp = spacy.load('en')

# Process a document, of any size
text = open('war_and_peace.txt').read()
doc = nlp(text)

# Hook in your own deep learning models
similarity_model = load_my_neural_network()
def install_similarity(doc):
    doc.user_hooks['similarity'] = similarity_model
nlp.pipeline.append(install_similarity)

doc1 = nlp(u'the fries were gross')
doc2 = nlp(u'worst fries ever')
doc1.similarity(doc2)
```

Features

- Non-destructive **tokenization**
- Syntax-driven sentence segmentation
- Pre-trained **word vectors**
- Part-of-speech tagging
- **Named entity** recognition
- Labelled dependency parsing
- Convenient string-to-int mapping
- Export to numpy data arrays
- GIL-free **multi-threading**
- Efficient binary serialization
- Easy **deep learning** integration
- Statistical models for **English** and **German**
- State-of-the-art speed
- Robust, rigorously evaluated accuracy

SEE EXAMPLES

SPACY IS TRUSTED BY

Quora Chartbeat DueDil STITCH FIX

wayblazer indico chattermill turi Kip

Socrata CYtora Signal N WONDERFLOW SYNAPSIFY

Preparing input for SVM filter: spaCy

```
import spacy

nlp = spacy.load(’en’)

def get_words():
    """ split story into tokens, lemmatize, remove junk,
         named-entities, and stop words """
    parsed_review = nlp(story)
    wlist = []
    for num, token in enumerate(parsed_review):
        if (len(token.lemma_) > 3) and \
           (token.lemma_.isalpha()) and \
           (token.ent_iob_ == ’O’) and \
            not (token.is_stop or token.is_punct or token.is_space or
                  token.like_num or token.is_oov):
            wlist.append(token.lemma_)
   return wlist
```

A sort of book on event data

A sort of book on event data:
Schrodt and Gerner 2000/2012 *Analyzing International Event Data*, chapts 1-3

A zillion papers:
`http://eventdata.parusanalytics.com/papers.dir/automated.html`.

If you like blogs:

`https://asecondmouse.wordpress.com/2014/02/14/the-legal-status-of-event-data/` (14 Feb 2014)

`https://asecondmouse.wordpress.com/2015/03/30/seven-observations-on-the-newly-released-icews-data/` (30 March 2015)

DARPA ICEWS 2007-2011 research phase

- Geographical focus: 27 countries in Asia with populations greater than 5-million
- Initially coded with open-source TABARI, which was then translated into Java as JABARI, with further enhancements to reduce false positives
- CAMEO ontology
- Factiva based for both major international sources and some local sources; density is 2000 to 4000 events per day
- Used to develop PITF-like forecasting models with PITF-like 80% accuracy

DARPA ICEWS 2012-present, operational phase

`https://asecondmouse.wordpress.com/2015/03/30/seven-observations-on-the-newly-released-icews-data/`

- Raytheon/BBN [proprietary] Serif/ACCENT coder
- Global coverage but events are still disproportionately from Asia
- Data release on Dataverse covers 1996-present with a rolling one-year embargo, released monthly
- BBN has extensively refined the CAMEO specification and coding manual is on Dataverse
- Includes very extensive actor dictionaries but not verb dictionaries
- Geolocated, though with quite a few errors

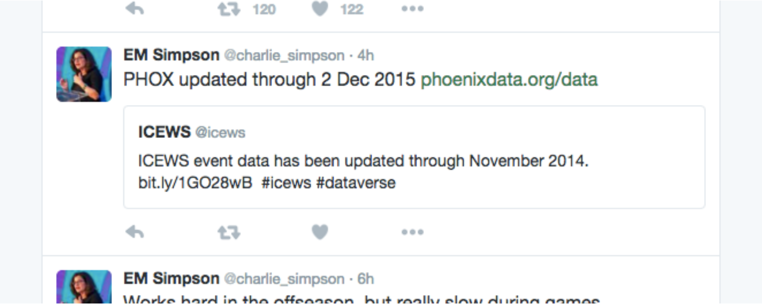
Converting ICEWS

Conversion program:

`https://github.com/philip-schrodt/text_to_CAMEO`

```
======= ICEWS original format =======

6826206 2004-01-01      Sudan           Sudan
Express intent to engage in diplomatic cooperation (such as policy support)
South Korea    South Korea     2603757 1   Korea Times  Sudan 15.5466 32.5336

6826211 2004-01-01      Recep Tayyip Erdogan
Sunni,Parties,Ideological,Center Right,Elite,International Religious
Turkey  Make statement  010     0
Justice and Development Party (National) Major Party,
Parties,Ideological,Center Right Turkey
2603789 2   Turkish Daily News   Ankara   Ankara  Turkey  39.9199 32.8543


======= Converted CAMEO format =======

2009-01-20   UZB  704  GOV  RUS  365  OTH  043     2.8     1
2009-08-27   CHN  710  GOV  TWN  713  OTH  042     1.9     1
```

But an issue with ICEWS remains:



EM Simpson @charlie_simpson · 4h

PHOX updated through 2 Dec 2015 phoenixdata.org/data

ICEWS @icews

ICEWS event data has been updated through November 2014. bit.ly/1GO28wB #icews #dataverse

EM Simpson @charlie_simpson · 6h

Works hard in the offseason, but really slow during games

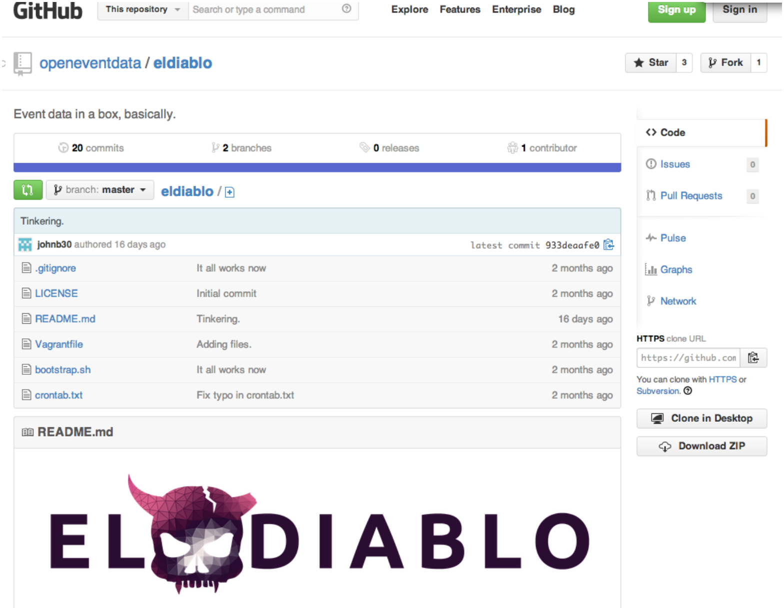
Source: Twitter at 1 pm. 3 December 2015

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

EL:DIABLO
Event Location: Dataset in a Box, Linux Option

- Full modular open-source pipeline to produce daily event data from web sources
- Scraper from white-list of RSS feeds and web pages
- Event coding from PETRARCH but other coders easily added to the pipeline
- Conventional one-a-day de-duplication keeping URLs of all duplicates
- Additional feature detectors are easily added
- Designed for implementation on Linux cloud servers



openeventdata / eldiablo

Event data in a box, basically.

20 commits | 2 branches | 0 releases | 1 contributor

eldiablo / +

Tinkering.

johnb30 authored 16 days ago latest commit 933deaafe0

.gitignore	It all works now	2 months ago
LICENSE	Initial commit	2 months ago
README.md	Tinkering.	16 days ago
Vagrantfile	Adding files.	2 months ago
bootstrap.sh	It all works now	2 months ago
crontab.txt	Fix typo in crontab.txt	2 months ago

README.md

HTTPS clone URL

`https://github.com`

You can clone with HTTPS or Subversion.

Phoenix Pipeline (Caerus Associates, 2014-2015)

mostly John Beieler
`https://github.com/openeventdata/phoenix_pipeline`
Andrew Halterman: Mordecai geolocation system
`https://github.com/caerusassociates/mordecai`

This near-real-time coding system is our successor to El Diablo. It has been producing data more or less continuously since summer 2014 at (`http://phoenixdata.org/`)

- Cloud-based
- White-list of RRS sources (currently about 300)
- Stanford CoreNLP + PETRARCH-1 coding
- One-a-day filtering
- Geolocation

Downside is that the system has dropped a few days due to unexpected crashes/reboots of the server where it is hosted: we are in the process of developing mirrors for it.

PETRARCH-1 (ca. Spring 2014)

Philip Schrodt and John Beieler
`https://github.com/openeventdata/petrarch`

- Written in Python, in contrast to the C++ TABARI
- Parsed input in the Penn Treebank format produced by Stanford Core NLP. This handles the noun/verb/adjective disambiguation that accounts for much of the size of the TABARI dictionaries
- Synonym sets from WordNet
- Identifies actors even if they are not in the dictionaries
- Extensive validation suite (about 250 cases)
- Codes at about 150 sentences per second, about a tenth the speed of TABARI but cluster computing is now readily available
- Problem: TABARI dictionaries—based on shallow parsing—do not always translate well to the higher precision provided by the Treebank parse

PETRARCH-2 (Caerus Associates, Summer 2015)

Clayton Norris
`https://github.com/openeventdata/petrarch2`

- Complete re-write of core event coding routines to use more of the information in the TreeBank parse
- Speed increased by roughly a factor of ten
- Verb dictionaries modified to work with the parse
- Additional debugging and robustness checks: in one recent test it was used to code a corpus of 25-million sentences from a variety of news sources and did not crash

NSF RIDIR Event Data Project

U.S. National Science Foundation Resource Implementations for Data Intensive Research in the Social Behavioral and Economic Sciences (RIDIR) Program: Modernizing Political Event Data for Big Data Social Science Research

- 3 years, currently about $2-million in total funding
- Lead institution: University of Texas at Dallas (Patrick Brandt).
- Other institutions include U of Oklahoma, U of Minnesota, U of Delaware, and John Jay College
- Roughly equal participation by political science and computer science departments
- Kickoff was early December 2015; still doesn't have a name, logo or t-shirts

RIDIR: Expansion of existing data sets

- Oklahoma negotiated a contract with Lexis-Nexis which allows them to download and code essentially the entire LN news archive: this should be finished by summer-2017
- Developing a multi-language coder to do native-language coding in English, Spanish and Arabic: work is currently underway
- "Containers" for deploying the system on large-scale parallel processing clusters for high volume and real-time coding
- Establishing "ground truth" validation sets in English, Spanish and Arabic covering all of the CAMEO/PLOVER categories
- NER systems for near-real-time updating of actors and open collaboration on maintenance of major actor dictionaries from Wikipedia, DBpedia, etc

Summary: once and future event data sources

- DARPA ICEWS: 1995-present (minus one year), updated monthly. Available on Dataverse.
- Open Event Data Alliance Phoenix: 2014-present, updated daily. `http://phoenixdata.org/`
- NSF RIDIR TERRIER (UT/Dallas, U of Oklahoma) [Temporally-Extended Reasonably Representative International Event Records]: Lexis-Nexis, 1980-2015.
- Cline Center (U of Illinois): *NY Times*, BBC Summary of World Broadcasts, FBIS: 1980 to present, with *NYT* extending back to 1945. Should be available in the very near future

Issues with CAMEO

- Almost all applications of CAMEO event data aggregated to either the 2-digit “cue category” or the even more general “quad category.”No one used all 260 codes.
- Nonetheless, users unfamiliar automated event coding sometimes assume every code had been equally well implemented.
- TABARI, PETRARCH-2 and ACCENT have implemented somewhat distinct “dialects” of CAMEO
- The complexity of CAMEO makes it almost impossible to generate a comprehensive set of “gold standard records” and human coders have difficulty agreeing on how to consistently distinguish many of the subcategories: this became particularly apparent as efforts were made to implement CAMEO in Spanish and Arabic.
- Newer coding systems provide information such as geolocation and named-entity extraction beyond the original date-source-target-event format and there was no standard for how to include these in the data.
- The continuing emphasis on coding substate activities demonstrated the need for either new categories or contexts to deal, for example, with criminal activity and events such as natural disaster, elections, and parliamentary behavior.

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

PLOVER objectives-1

- 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.
- The CAMEO 01 and 02 categories dealing with comments have been eliminated.
- The CAMEO 08 “YIELD” category has been split into verbal (CONCEDE) and material (RETREAT) components.
- CAMEO categories 18, 19, 20 dealing with violence are combined into a single ASSAULT category .
- A new category has been added for criminal behavior.
- The complexity of substate actor codes has been limited, and the allowable substate modifiers have been substantially simplified.

PLOVER objectives-2

- 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: COERCE modes

Name	Content
confiscate	confiscate property
destroy	destroy property
restrict	impose restrictions on political freedoms or movement
ban	ban individuals or organizations
censor	censor, ban or restrict access to publications
curfew	impose curfew
martial-law	impose state of emergency or martial law
arrest	arrest, detain, or charge with legal action
deport	expel or deport individuals

Adapted from CAMEO category 17x

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 JSON

Name	Content	Note	Required?
id	unique identifier	1	Y
date	date in YYYY-MM-DD format		Y
time	ISO 8601-formatted time	2	N
enddate	date in YYYY-MM-DD format		Y
endtime	ISO 8601-formatted time	2	N
source	list of actor objects		Y
target	list of actor objects		N
event	event category		Y
eventLoc	location object for event		N
eventText	text of event		N
quadCode	1, 2, 3 or 4		N
eventScale	floating point scale value		N
mode	mode category	3	N
context	context category	3	N
dead	number killed		N
injured	number injured		N
size	number: depends on context		N
link	link identifier	4	N
text	text from which the record was coded		N
citation	bibliographic citation or database identifier for text		N
url	URL for text		N
language	language of text (ISO 639-1 two-letter codes)		N
publication	name of text publisher		N
license	license covering text		N
copyright	copyright covering text		N
textInfo	textInfo object for text		N
coder	coder identification		N
version	version of data set		N
codebook	reference for the codebook used to code the text		N
dateCoded	date of coding		N
comment	any text		N

PLOVER: Actor JSON

Table: Information object for actors

Name	Content
code	3-char actor code
sector	3- or 6-char source sector
identifer	unique identifier for source [see Note 1]
actorLoc	location object
actorText	extracted text for source
religion	religion (code or text)
ethnicity	ethnicity (code or text)
office	office or official position (code or text)
gender	gender (code or text)
age	integer

Notes:

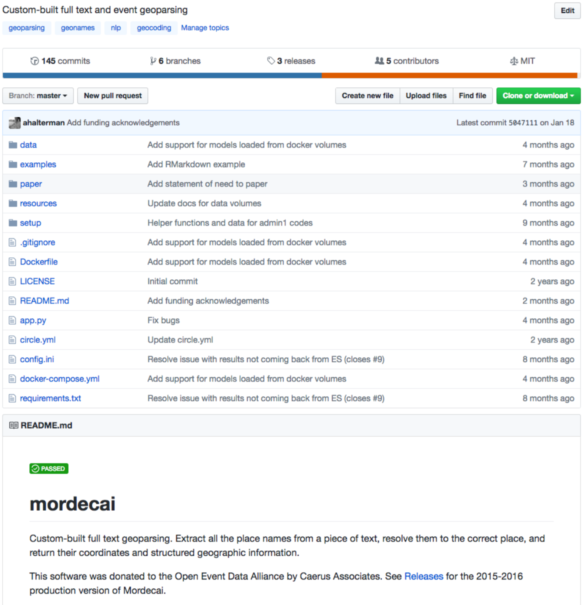
1. These fields would be used to resolve the name of an actor that occurs in multiple forms—for example "Islamic State", "IS", "ISIS", "Daesh"—into a single form or code (for example the organization number in the TORG typology).

Open question: How to define an event data coding scheme?

- Codebook: all human-coded datasets, beginning with WEIS and COPDAB
- Dictionaries/patterns: this is effectively how CAMEO is defined, since it is implemented in automated coders such as TABARI, PETRARCH and Serif/ACCENT
- Examples: this would be best for future machine-learning systems, but large sets of examples are expensive to generate

The LDC Gigaword news story corpus (2000-2010) would provide a generally accessible set of example cases that are very representative of event data sources.

Mordecai geolocation



Custom-built full text and event geoparsing

ahalterman Add funding acknowledgements — Latest commit 5047111 on Jan 18

data	Add support for models loaded from docker volumes	4 months ago
examples	Add RMarkdown example	7 months ago
paper	Add statement of need to paper	3 months ago
resources	Update docs for data volumes	4 months ago
setup	Helper functions and data for admin1 codes	9 months ago
.gitignore	Add support for models loaded from docker volumes	4 months ago
Dockerfile	Add support for models loaded from docker volumes	4 months ago
LICENSE	Initial commit	2 years ago
README.md	Add funding acknowledgements	2 months ago
app.py	Fix bugs	4 months ago
circle.yml	Update circle.yml	2 years ago
config.ini	Resolve issue with results not coming back from ES (closes #9)	8 months ago
docker-compose.yml	Add support for models loaded from docker volumes	4 months ago
requirements.txt	Resolve issue with results not coming back from ES (closes #9)	8 months ago

mordecai

Custom-built full text geoparsing. Extract all the place names from a piece of text, resolve them to the correct place, and return their coordinates and structured geographic information.

This software was donated to the Open Event Data Alliance by Caerus Associates. See Releases for the 2015-2016 production version of Mordecai.

Using Mordecai

Mordecai is most easily run as a Docker container which involves, well, Docker. Which mostly works most of the time.

```
INPUT:
curl -XPOST -H "Content-Type: application/json"  --data ’{"text":"(Reuters) -
The Iraqi government claimed victory over Islamic State insurgents in Tikrit
on Wednesday after a month-long battle for the city supported by Shiite
militiamen and U.S.-led air strikes, saying that only small pockets of resistance
<...rest of story...>
and American support to get back on its feet.", "country": "IRQ"}’
 ’http://localhost:5000/places’
```

```
OUTPUT:
[{"lat": 34.61581, "placename": "Tikrit", "seachterm": "Tikrit", "lon": 43.67861,
"countrycode": "IRQ"}, {"lat": 34.61581, "placename": "Tikrit", "seachterm": "Tikrit",
"lon": 43.67861, "countrycode": "IRQ"}, {"lat": 33.32475, "placename": "Baghdad",
"seachterm": "Baghdad", "lon": 44.42129, "countrycode": "IRQ"}]
```

TRIGGER WARNING: A diversionary discourse on why you should get some experience with computer programming.

Two data science fundamentals

1. Because of volume-velocity-variety nature of “big data”, most data science projects involve a great deal of time simply getting the information into the form where it can be analyzed. This probably comes close to an 80/20 ratio.

2. Probably a majority of the software, and some of the methodology, you will be using in ten years does not exist today: you absolutely must be able to continually learn and adapt. But the good news

- Everything is open source
- The available on-line support is incredible
- Some fundamentals will not change: SVM and logit are still “embarassingly effective”; Unix is still Unix

What, exactly, is “programming”?

Professionally, the “IT” field is now sub-divided into an astonishing number of niches, up to and including “Scrum Master” (Google it...). These are largely for the benefit of managers who, from the very dawn of programming, hate programmers because skilled programmers don’t respond to the conventional management tools of fear, greed and delusion. I digress.

Contemporary programming largely involves the following skills :

- Constructing code involving loops and conditionals
- Knowing how to apply data structures: Python has about half a dozen that are used commonly
- Ability to quickly grok and apply libraries: Python and Java are notably library-rich
- Debugging, which is always about half the job and is a *general* still
- Doing these tasks in several, but not too many, programming languages

Why Python?

- Stable and standardized across platforms and widely available/documented; massive and reasonably civil user community. Several very good MOOCs.
- Core language is quite simple (arguably, too simple. . .)
- Automatic memory management (unlike C/C++)
- Text oriented rather than GUI oriented (unlike Java). More coherent than perl, particularly when dealing with large programs
- Extensive libraries but these are optional (unlike Java) and you can do a lot with very small subsets of the language
- C code can be easily integrated using “python” for high-performance applications
- There are Python libraries for all commonly used data management, statistical and machine learning approaches: Python can replace R in most analyses

Besides...

But the main reason you need Python (or Java):

Besides...

But the main reason you need Python (or Java):

Pipeline and glue programs!

Besides...

But the main reason you need Python (or Java):

Pipeline and glue programs!

(which are *possible*, but not particularly practical, in *R*)

And if all else fails. . .

Mainstream political science methodology training supplemented with data science training in machine learning and visualization plus a basic competence in Python and/or *R* is a nearly ideal combination for someone intending to work with human-generated “big data.”

It is, in fact, far better training than most computer science programs provide because CS publications generally use standardized data sets in order to generate relative performance metrics. “Big data” inputs for political science, in contrast, are usually a complete mess: learning to deal with messy real-world input is good!

Meanwhile the most recent APSA employment report—hey, they’re saying this, not me—notes only about a third of political science PhDs can expect to get a tenure-track position.

Demand for data scientists, on the other hand, is likely to outpace supply for at least the next decade.

Bonus: These positions do not involve grading blue books or lecturing on controversial political issues to young adults carrying concealed firearms.

End of rant: we now return you to the advertised topic of this webinar

Event data coding programs

- TABARI: C/C++ using internal shallow parsing.
  `http://eventdata.parusanalytics.com/software.dir/tabari.html`

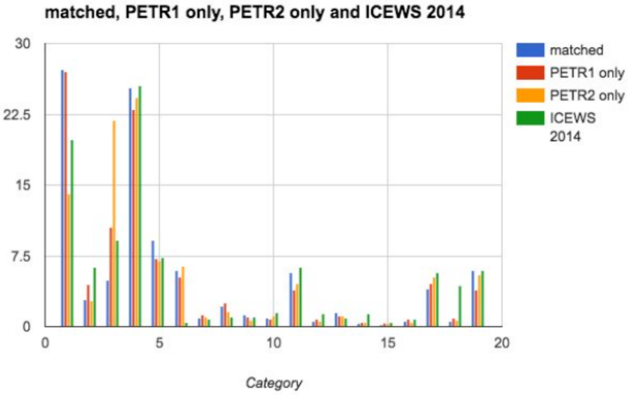
- JABARI: Java version of TABARI with additional enhancements: alas, abandoned and lost following end of ICEWS research phase

- 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

Clarification in response to question in talk:

You can develop a customized coding system simply by modifying the open source dictionaries used by these existing coders; you don't need to write your own program

- In all of the programs, the actor codings are determined entirely by the dictionaries, which are open
- Because the default dictionaries were designed for global coding, they do not have a lot of local detail, for example on small militant and/or opposition groups and/or criminal gangs. Adding these should be fairly straightforward, just a day or two to get the common actors.
- TABARI and PETRARCH-1 also get all of their information on *event* codings from the dictionaries as well; PETRARCH-2 and ACCENT, in contrast, have some aspects of their dialects of CAMEO hard-coded into the program
- Due to intellectual property constraints, you still need to acquire source texts, though if you are looking at a limited geographical area and time frame, this may not be difficult.

So, punk, ya think ya can write
an event coder?

This is no longer completely insane…

- In all likelihood, the generic event coders such as PETRARCH and ACCENT are collecting both more and less information than you want: they are, well, generic
- Stories within a limited domain, such as protests, are structured more specifically than stories in a general domain (e.g. everything ever posted on Lexis-Nexis)
- The availability of tools from the computational linguistics community means that much of the complexity can be handled through pre-processing (or, in the case of geolocation, post-processing)
- The recently developed "universal dependency parse" seems particularly attractive

Note also that in recent years Javier Osorio (Political Science, Notre Dame), Alex Hanna (Sociology, Wisconsin) and John Beieler (Political Science, Penn State) all developed automated coding systems as part of their dissertation research.

Besides...

The shortest answer is doing.
Ernest Hemingway

Besides...

The shortest answer is doing.
Ernest Hemingway

Do or do not: there is no “try.”

Besides...

The shortest answer is doing.
Ernest Hemingway

Do or do not: there is no “try.”
Yoda

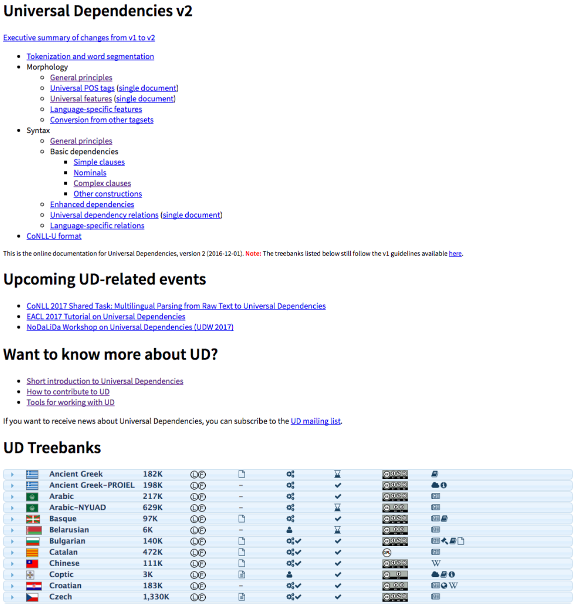
Besides...

The shortest answer is doing.
Ernest Hemingway

Do or do not: there is no “try.”
Yoda

Yeah, I can do that. . .
Miles Walsh

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?

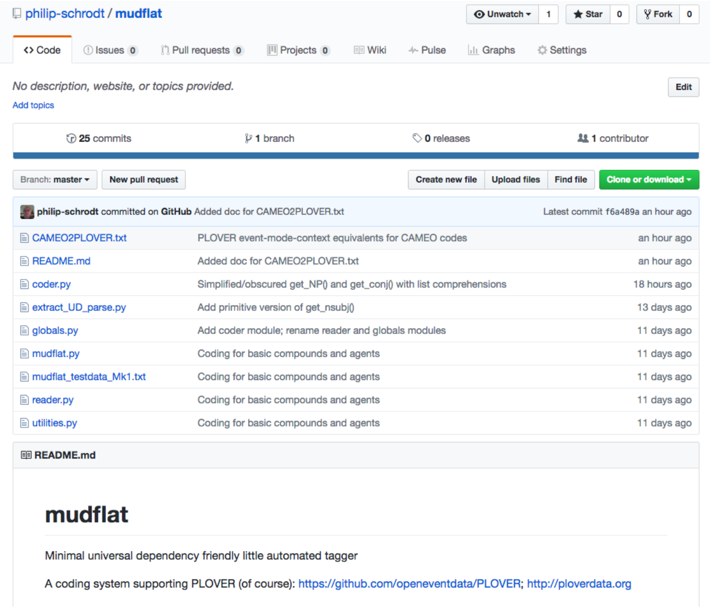
- Short introduction to Universal Dependencies
- How to contribute to UD
- Tools for working with UD

If you want to receive news about Universal Dependencies, you can subscribe to the UD mailing list.

UD Treebanks

Ancient Greek	182K	ⓁⒻ	🗋	⚙	⌛	cc BY NC SA	📘
Ancient Greek-PROIEL	198K	ⓁⒻ	–	⚙	✓	cc BY NC SA	☁ ℹ
Arabic	217K	ⓁⒻ	–	⚙	✓	cc BY NC SA	📰
Arabic-NYUAD	629K	ⓁⒻ	–	⚙	⌛	cc BY SA	📰
Basque	97K	ⓁⒻ	🗋	⚙	✓	cc BY NC SA	📰 📘
Belarusian	6K	ⓁⒻ	–	👤	⌛	cc BY SA	📰
Bulgarian	140K	ⓁⒻ	🗋	⚙✓	✓	cc BY NC SA	📰 🔨 📘 🗋
Catalan	472K	ⓁⒻ	🗋	⚙✓	✓	GPL	📰
Chinese	111K	ⓁⒻ	🗋	⚙✓	✓	cc BY NC SA	W
Coptic	3K	ⓁⒻ	🗎	👤	✓	cc BY	☁ 📘 ℹ
Croatian	183K	ⓁⒻ	–	⚙✓	✓	cc BY SA	📰 🌐 W
Czech	1,330K	ⓁⒻ	🗎	⚙✓	✓	cc BY NC SA	📰

So. . .



No description, website, or topics provided.

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

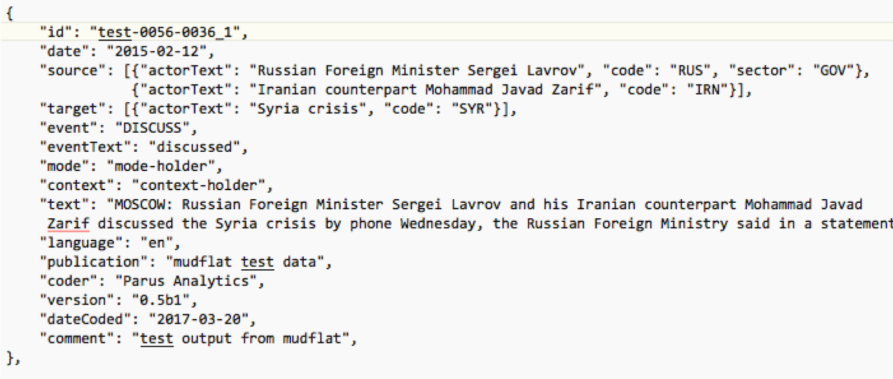
README.md

mudflat

Minimal universal dependency friendly little automated tagger

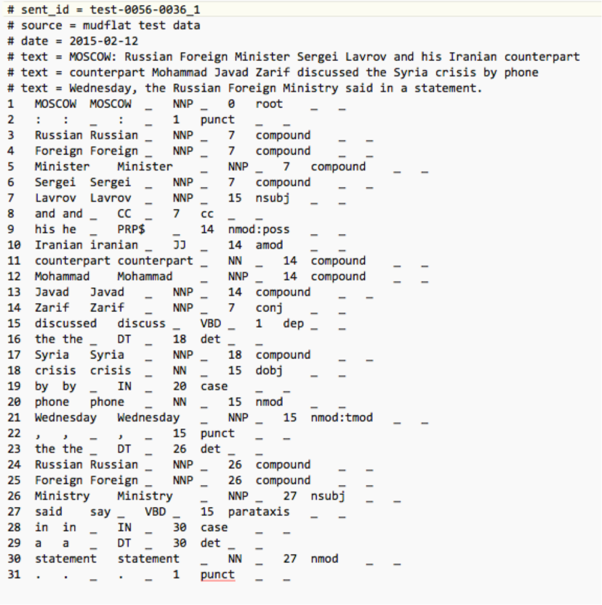
A coding system supporting PLOVER (of course): https://github.com/openeventdata/PLOVER; http://ploverdata.org

PLOVER output



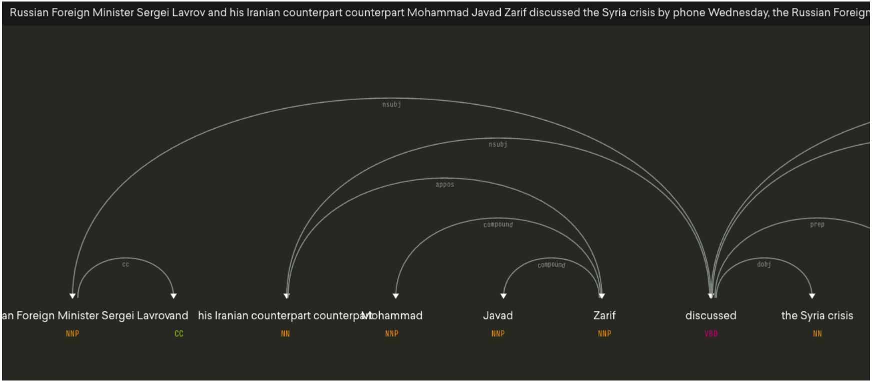
```
{
    "id": "test-0056-0036_1",
    "date": "2015-02-12",
    "source": [{"actorText": "Russian Foreign Minister Sergei Lavrov", "code": "RUS", "sector": "GOV"},
               {"actorText": "Iranian counterpart Mohammad Javad Zarif", "code": "IRN"}],
    "target": [{"actorText": "Syria crisis", "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",
},
```

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   _   _
```

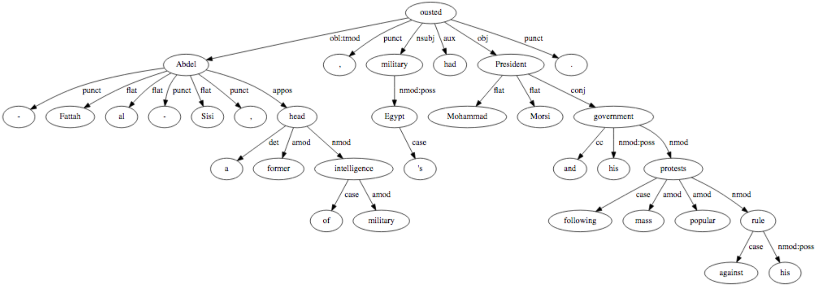
Visualization: displaCy



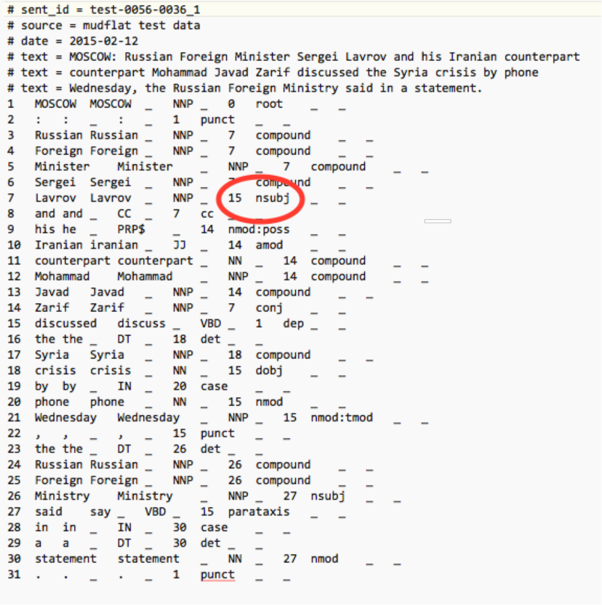
Russian Foreign Minister Sergei Lavrov and his Iranian counterpart counterpart Mohammad Javad Zarif discussed the Syria crisis by phone Wednesday, the Russian Foreign

`https://demos.explosion.ai/displacy/`

Visualization: TensorFlow

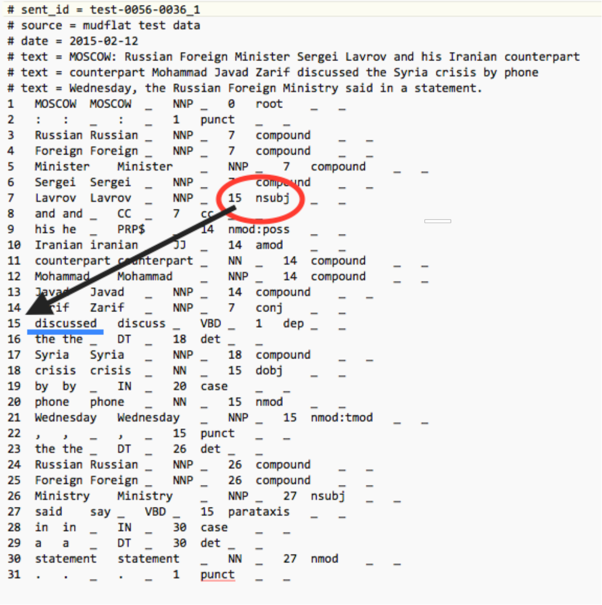


Dependency parse: locate subject



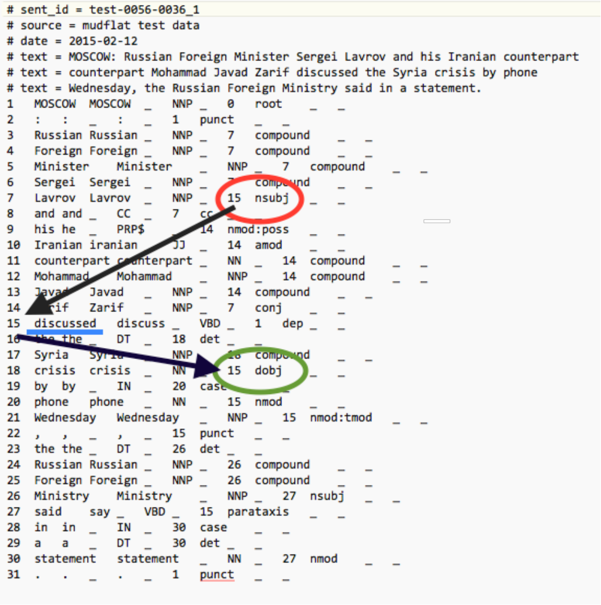
```
# sent_id = test-0056-0036_1
# source = mudflat test data
# date = 2015-02-12
# text = MOSCOW: Russian Foreign Minister Sergei Lavrov and his Iranian counterpart
# text = counterpart Mohammad Javad Zarif discussed the Syria crisis by phone
# text = Wednesday, the Russian Foreign Ministry said in 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



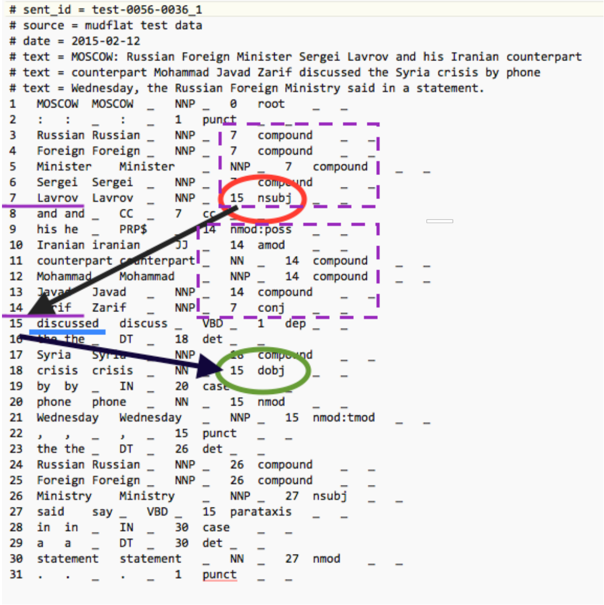
```
# sent_id = test-0056-0036_1
# source = mudflat test data
# date = 2015-02-12
# text = MOSCOW: Russian Foreign Minister Sergei Lavrov and his Iranian counterpart
# text = counterpart Mohammad Javad Zarif discussed the Syria crisis by phone
# text = Wednesday, the Russian Foreign Ministry said in 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



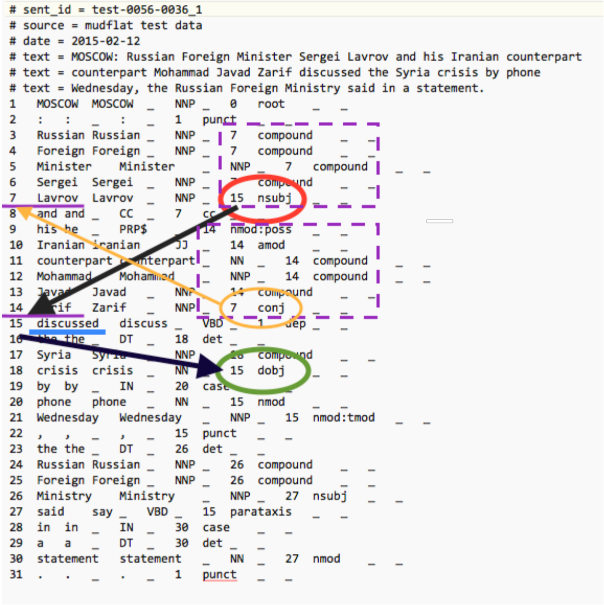
```
# sent_id = test-0056-0036_1
# source = mudflat test data
# date = 2015-02-12
# text = MOSCOW: Russian Foreign Minister Sergei Lavrov and his Iranian counterpart
# text = counterpart Mohammad Javad Zarif discussed the Syria crisis by phone
# text = Wednesday, the Russian Foreign Ministry said in 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   _   _
```

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   _   _
```

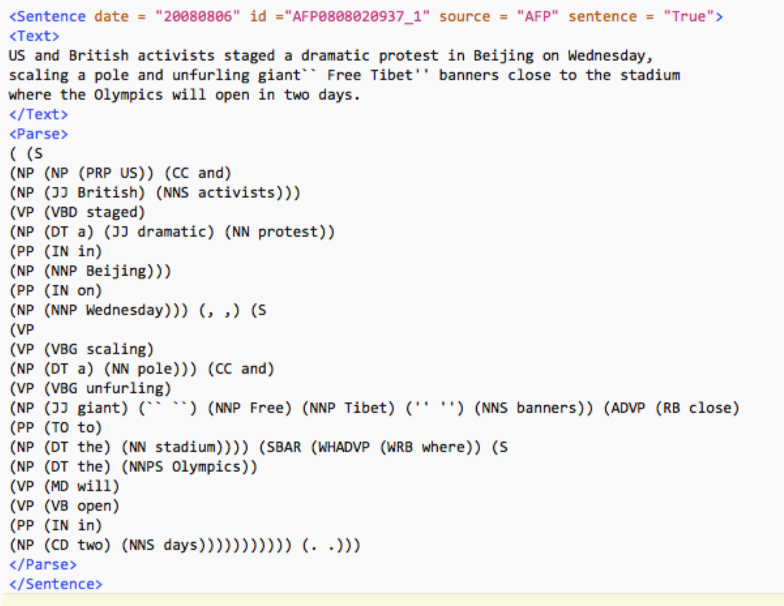
Main event coding: mudflat

```
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
```

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 """
    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
```

PETRARCH input: constituency parse tree



```
<Sentence date = "20080806" id ="AFP0808020937_1" source = "AFP" sentence = "True">
<Text>
US and British activists staged a dramatic protest in Beijing on Wednesday,
scaling a pole and unfurling giant`` Free Tibet'' banners close to the stadium
where the Olympics will open in two days.
</Text>
<Parse>
( (S
(NP (NP (PRP US)) (CC and)
(NP (JJ British) (NNS activists)))
(VP (VBD staged)
(NP (DT a) (JJ dramatic) (NN protest))
(PP (IN in)
(NP (NNP Beijing)))
(PP (IN on)
(NP (NNP Wednesday))) (, ,) (S
(VP
(VP (VBG scaling)
(NP (DT a) (NN pole))) (CC and)
(VP (VBG unfurling)
(NP (JJ giant) (`` ``) (NNP Free) (NNP Tibet) ('' '') (NNS banners)) (ADVP (RB close)
(PP (TO to)
(NP (DT the) (NN stadium)))) (SBAR (WHADVP (WRB where)) (S
(NP (DT the) (NNPS Olympics))
(VP (MD will)
(VP (VB open)
(PP (IN in)
(NP (CD two) (NNS days)))))))))) (. .)))
</Parse>
</Sentence>
```

Code event coding: PETRARCH-2

```
def get_meaning(self):
    self.get_meaning = self.return_meaning
    c, passive,meta = self.get_code()
    if c:
        curparse = ’==CODED==’
    else:
        curparse = self.get_parse_string()

    s_options = filter(lambda a: a.label in "SBAR",self.children)

    def resolve_events(event):
        returns = []
        first,second,third = [up,"",""]
        if not (up or c) :
            return [event]
        if not isinstance(event,tuple):
            second = event
            third = c
            if passive:
                for item in first:
                    e2 = ([second],item,passive)
                    self.sentence.metadata[id(e2)] = [event,meta,7]
                    returns.append(e2)
        elif event[1] == ’passive’:
            first = event[0]
            third = utilities.combine_code(c,event[2])
            if up:
                returns = []
                for source in up:
                    e = (first,source,third)
                    self.sentence.metadata[id(e)] = [event,up,1]
                    returns.append(e)
                return returns
            second = ’passive’
```

Code event coding: PETRARCH-2

```
    elif not event[0] in ['',[],[""],["~"],["~~"]]:
        second = event
        third = c
    else:
        second = event[1]
        third = utilities.combine_code(c,event[2])
    e = (first,second,third)
    self.sentence.metadata[id(e)] = [event,c,meta ,2]
    return returns + [e]

events = []
up = self.get_upper()
if self.check_passive() or (passive and not c):
    # Check for source in preps
    source_options = []
    target_options = up
    for child in self.children:
        if isinstance(child,PrepPhrase):
            if child.get_prep() in ["BY","FROM","IN"]:
                source_options += child.get_meaning()
                meta.append((child.prep, child.get_meaning()))
            elif child.get_prep() in ["AT","AGAINST","INTO","TOWARDS"]:
                target_options += child.get_meaning()
                meta.append((child.prep, child.get_meaning()))
    if not target_options:
        target_options = ["passive"]
    if source_options or c:

        for i in target_options:
            e = (source_options, i ,c if self.check_passive() else passive)
            events.append(e)
            self.sentence.metadata[id(e)] = [None,e,meta,3]
            self.meaning = events
            return events
```

Code event coding: PETRARCH-2

```
up = "" if up in [’’,[],[""],["~"],["~~"]] else up
low,neg = self.get_lower()
if not low:
    low = ""
if neg:
    c = 0

if isinstance(low,list):
    for event in low:
        events += resolve_events(event)
elif not s_options:
    if up or c:
        e = (up,low,c)
        self.sentence.metadata[id(e)] = [None,e,4]
        events.append(e)
    elif low:
        events.append(low)

lower = map(lambda a: a.get_meaning(),s_options)
sents = []

for item in lower:
    sents += item

if sents and not events:  # Only if nothing else has been found do we look at lower NP’s?
                          # This decreases our coding frequency, but removes many false positives
    for event in sents:
        if isinstance(event,tuple) and (event[1] or event[2]):
            for ev in resolve_events(event):
                if isinstance(ev[1],list):
                    for item in ev[1]:
                        local = (ev[0],item,ev[2])
                        self.sentence.metadata[id(local)] = [ev,item,5]
                        events.append(local)
```

Code event coding: PETRARCH-2

```
                else:
                    events += resolve_events(event)
if events and isinstance(events[0],tuple):
    if events[0][0] and events[0][1] and not events[0][2]:
        utilities.nulllist.append((curparse, events[0]))

maps = []
for i in events:
    evs = self.match_transform(i)
    if isinstance(evs,tuple):
        for j in evs[0]:
            maps.append(j)
            self.sentence.metadata[id(j)] = [i,evs[1],6]
    else:
        maps += evs
self.meaning = maps
return maps
```

Additional work to be done

Some future challenges/opportunities - 1

- Customized coding of the TERRIER corpus
- Common actor dictionary—including non-state actors—with historical coverage and near-real-time updates from news, European Media Monitor, Wikipedia and DBpedia
- Extensive set of "gold standard records" based on Gigaword or some other shared corpus
- Rapid dictionary development methods for languages beyond Spanish and Arabic
- Robust mirroring of Phoenix (or equivalent) data

Some future challenges/opportunities - 2

- Methods for calibrating the long time series to account for changes in the news environment
- Creative applications of near-real-time data
- Specialized, and more detail, behavior-specific data sets: "protest" seems to be the one most in demand
- Continued refinement of geolocation definitions (not everything has a location. . .) and methods

Thank you

Email:
schrodt735@gmail.com

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

Links to data and software: `http://philipschrodt.org`

Blog: `http://asecondmouse.org`