Three's a Charm?:

Open Event Data Coding with EL:DIABLO, PETRARCH, and the Open Event Data Alliance.*

Philip A. Schrodt [schrodt735@gmail.com]

John Beieler [john.b30@gmail.com]

Muhammed Idris [muhammedy.idris@gmail.com]

Version 1.0: March 23, 2014

*Paper presented at the International Studies Association meetings, Toronto, March 2014. The authors would like to acknowledge helpful assistance and comments from Patrick Brandt, Andrew Halterman and Erin Simpson, as well as support from the U.S. National Science Foundation Political Science and Methods, Measurements and Statistics programs, award SES-1259190, and the Penn State "Big Data Social Science" IGERT program, funded by the U.S. National Science Foundation award DGE-1144860. Affiliations for the authors: Schrodt : Parus Analytical Systems, State College, PA 16803. Beieler and Idris: Department of Political Science, Pennsylvania State University, University Park, PA 16802 USA. This paper is available at `http://eventdata.parusanalytics.com/papers.dir/automated.html`. The open source software for this system can be found at `https://github.com/openeventdata`.

Abstract

This paper is a brief review of three current efforts to provide an open and transparent path to the automated production of event data:

- EL:DIABLO: an open, user-friendly modular system for the acquisition and coding of web-based news sources which is intended to allow small research teams to generate customized event data sets with a minimum of effort
- PETRARCH: a Python-based event data coder using fully-parsed Penn Treebank input
- The Open Event Data Alliance, a new professional organization for the promotion and provision of fully transparent open event data

All truth passes through three stages. First, it is ridiculed. Second, it is violently opposed. Third, it is accepted as being self-evident.
Arthur Schopenhauer

1 Introduction

Political event data have long been used in the quantitative study of international politics, dating back to the early efforts of Edward Azar's COPDAB [Azar, 1980] and Charles McClelland's WEIS [McClelland, 1976], as well as a variety of more specialized efforts such as Leng's BCOW [Leng, 1987]. The 1990s saw the development of two practical automated event data coding systems: the NSF-funded KEDS/TABARI (`http://eventdata.parusanalytics.edu`; Gerner et al. [1994], Schrodt and Gerner [1994], Schrodt [2006]) and the proprietary VRA-Reader (`http://vranet.com`; King and Lowe [2004])[1] and in the 2000s, two new political event coding ontologies–CAMEO [Schrodt et al., 2009] and IDEA [Bond et al., 2003]–were designed for implementation in automated coding systems. A summary of the current status of political event projects can be found in [Schrodt, 2012b, Bernauer and Gleditsch, 2012].

For the past two decades, event data analysis has been driven—in some cases very dramatically—by two exponential changes: the increase in machine-readable text (now available more or less for free), and the increase in computing power that provides for the efficient processing of that text. Humans are able to code about six to ten events per hour; TABARI codes on a single processor at about 2,000 events per second—itself an increase by a factor of about a million over human coding—and through simple parallel processing this can be scaled indefinitely with a near linear increase in speed.

Over the past two or three years, these developments were further accelerated by the social and technological revolutions driving highly successful open collaborative research environments such as Linux, R and Python, which have allowed decentralized collaborations with diffuse funding to match or exceed the efforts of multi-million-dollar centralized efforts. The increasingly widespread low-cost "cloud" computing resources allow small groups to temporarily access massively parallel computing facilities without the need to establish and maintain these when they are not in use, and increasingly standardized computing environments, particularly the Unix/Linux environments commonly used in research computing, enable researchers to seamlessly transfer both projects and software across systems.

[1]The BBN Technologies SERIF system has apparently recently been adapted to do event coding, but all information on this appears to be restricted, and in any case neither the system nor data are available to the open research community.

Since the last ISA meeting, the field of event data has gone through a series of highs and lows which, on the instructions of legal counsel, cannot be addressed by individuals affiliated with Penn State, which includes two of the authors of this paper (see, however, Schrodt [2014]). While the eventual resolution of those issues is unclear, what has become clear is that the collective event data enterprise has reached a point where it can only function effectively if data can be produced with as much transparency as is possible given the intellectual property constraints which apply to the source texts[2] and, as we argue below, event data has reached a point where to operate effectively it needs some formal institutional structure comparable to that found in a large number of other research communities that have developed around open source software.

This paper is a brief review of three current efforts to provide an open and transparent path to the automated production of event data that involves three major components directed towards these issues:

- EL:DIABLO: an open, user-friendly modular system for the acquisition and coding of web-based news sources which is intended to allow small research teams to generate customized event data sets with a minimum of effort;
- PETRARCH: a Python-based event data coder using fully-parsed Penn Treebank input;
- The Open Event Data Alliance, a new professional organization for the promotion and provision of fully transparent open event data.

This list is not by any means exhaustive, nor will we be providing equal levels of detail on all components of the project. For example, a major feature of EL:DIABLO is geocoding software being developed by the GeoVista project at Penn State (`http://www.geovista.psu.edu/`) which receives funding by organizations such as the National Geospatial Intelligence Agency, and additional efforts are underway at the University of Texas at Dallas (servers; parallel processing resources; additional geocoding software), the University of Minnesota (mirroring servers) and Caerus Analytics (white-list source development; issues coding). These efforts are moving rapidly and this paper is likely to be outdated by the time most people are reading it; more current updates will be found at the various web sites referenced throughout the paper, but particularly `https://github.com/openeventdata` and `http://openeventdata.org` for the institutional developments.

[2]see `http://asecondmouse.wordpress.com/2014/02/14/the-legal-status-of-event-data/` for a discussion of these issues. While again reminding the dear reader that Schrodt is not a lawyer, that post has received nearly 1,000 views, presumably some from individuals who are lawyers, and no one has suggested modifying it.

EL:DIABLO

EL:DIABLO [Event/Location: Dataset In A Box, Linux-Option; ELDI] is a transparent, user-friendly, integrated system for the generation of real-time event data from a white-list of RSS feeds and web pages. ELDI is instantiated as a Vagrant box, which allows for the deployment of the system on any operating system and hardware configuration.[3] The setup can also be installed, with very little modification, on any Linux installation: it was developed on a Linode Ubuntu installation that cost us $20/month. More information about EL:DIABLO itself is available at `http://openeventdata.github.io/eldiablo/`.

In contrast to earlier systems which contained a single processing pipeline, ELDI is designed—on Unix principles, and paralleling the open source developments of R and Python—to be highly modular and readily customized by advanced users (as well as allowing the rapid incorporation of new components). At the same time, however, it is intended to be readily useable by practitioners without extensive knowledge of computer science and Big Data: it is intended to encourage a proliferation of specialized data sets (following the examples of survey research, protest and composite conflict data) rather than a single "one dataset to rule them all" approach. ELDI is completely open and available on GitHub: `https://github.com/openeventdata`.

ELDI currently involves the following discussed below: these have all been operational for about a month and a fully operational version is currently posted on the GitHub site but we still consider it to be in beta development.[4]

1.1 Web scraper

In order to obtain the news stories that serve as the base for the coded event data ELDI makes use of the robust Python ecosystem. Much of the heavy lifting for scraping the web pages is done by the `Goose` (`https://github.com/grangier/python-goose`) library, which allows for scraping of arbitrary pages without purpose-made scrapers. Rather than using a vacuum-style approach to obtaining news articles, a whitelist of RSS feeds is utilized to better pinpoint relevant news stories. ELDI currently has 166 unique RSS feeds from which

[3] Vagrant eases the use of virtual machines on any operating system. Vagrant is commonly used in development operations and software engineering to overcome "dependency hell," which refers to the difficulty of ensuring that software runs in the exponential number of possible setups.

[4] In contrast to most software projects, development is taking longer and is proving to be a bit more complicated than we originally anticipated...

stories are scraped. Each of the scraped stories is stored in a MongoDB database, along with the relevant metadata, for easy access at future dates.

1.2 Full-story filter

New story sources have always contained a large number of stories which do not contain political events. Sports stories are the most notorious category due to the ubiquitous tendency of sports writers to use elaborate military metaphors to inflate the importance of adults engaged in meaningless games whose primary purpose is providing a venue for beer advertisements: for example in the *Gigaword* corpus about a quarter to a third of stories are sports. The TABARI dictionaries contained a extensive list of sports-related vocabulary—which is usually quite distinctive—and was fairly effective at getting rid of these.[5]

There are, however, a large number of other categories—though not nearly as large a number of stories—which ideally should also be removed from the text stream. With the rise of cable channels which treat business competition as a form of entertainment, one now sees bland business transactions conducted over conference calls described with vocabulary reserved in earlier times for describing the titanic clash of vast armies. Time-shifted events are also problematic, particularly when these occur in the distant past: the vocabulary for short shifts in time ("last week", "on Tuesday") is relatively easy to deal with, and there is a fairly well developed technology, including the "TimeML" markup language (`http://timeml.org/site/index.html`) for dealing with this. More distant times, as well as routine announcements of anniversaries and calendars of events, however, can be more problematic, particularly for coding systems expecting subject-verb-object sentences.

We are currently in the process of experimenting with standard supervised text classification algorithms—notably support vector machines—to deal with this issue. Assuming this approach works, these will be straightforward to implement because these algorithms are simply trained by example, requiring only a set of positive and negative cases, rather than a detailed specification. We anticipate a series of these classifiers, implemented in either *R* or Python, which can optionally be included in the processing pipeline.

[5]Objections are typically raised that this will also eliminate sports-related international incidents such as the July 1969 "Football War"/"Soccer War" between El Salvador and Honduras: `http://en.wikipedia.org/wiki/Football_War`. This is a precision/specificity (false positive/negative) trade-off, there's no free lunch, and you just have to make a choice. In this instance, miss one true positive over a period of fifty years versus pick up about a hundred and fifty false positives on a typical Sunday afternoon, and the choice is probably evident.

1.3 Mongo-formatter

This component segments the sentences that have been extracted and handles the basic formatting for later input to the other components of the pipeline, specifically producing the XML-based PETRARCH format shown in Figure 1. At the present time, we are using a couple of fairly general formatters, but in the future anticipate writing customized formatters—particularly at the initial level of extracting the natural-language texts from the scraped HTML files—for high-priority sources such as the major international news agencies. This is another point where incremental developments from a large open community may be useful: formatters are an excellent classroom exercise for learning perl or Python.

1.4 Stanford Core NLP parser [SCNLP]

This is the most popular parser[6] and it being used to produce the input to both PETRARCH and the Penn State GeoData geolocation program, but any parser which can produce a Penn Treebank-formatted parse tree (`http://www.cis.upenn.edu/~treebank/`; see example in Figure 1) can be substituted[7]

The Treebank output is only one of a number of mark-ups provided by the SCNLP system—Figure 2 shows a number of the other options—and it is quite possible that in the relatively near future we will be able to exploit some of these other features as well. The key distinction here is that with the increased availability of open-source solutions, we are now moving away from the specialized parsers used in KEDS, TABARI, the VRA-Coder and presumably BBN-SERIF to general solutions from the much larger NLP community.

1.5 Event Coding

Until something better comes along, we will be using the PETRARCH coder, described in Section 2 as the primary event coder. However, this is a substitutable component like any other.

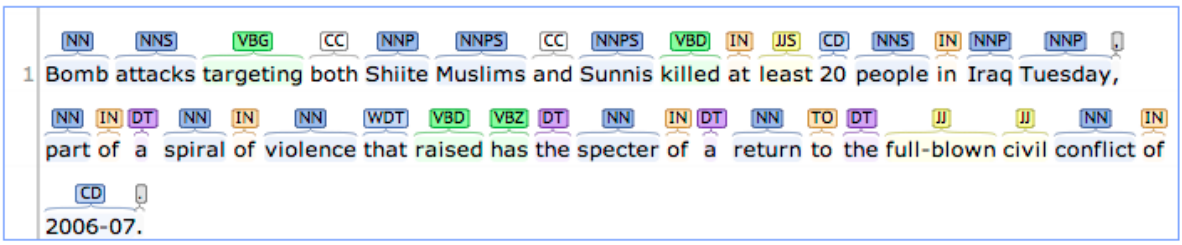
[6]The extensive processing pipeline in the GATE system—`https://gate.ac.uk/overview.html`—is the other obvious alternative.

[7]A distinct downside to the Stanford system is that it is the only component of ELDI that is not available in Python: it is a Java program. The Python equivalent, the *nltk* library, meanwhile seems primarily designed for experimentation and toy problems (SCNLP is certainly not confined to toy problems: witness the coding of the Gigaword corpus). In an ideal world, someone would write the equivalent of SCNLP in Python, and were such a system to become available, we would probably substitute it.

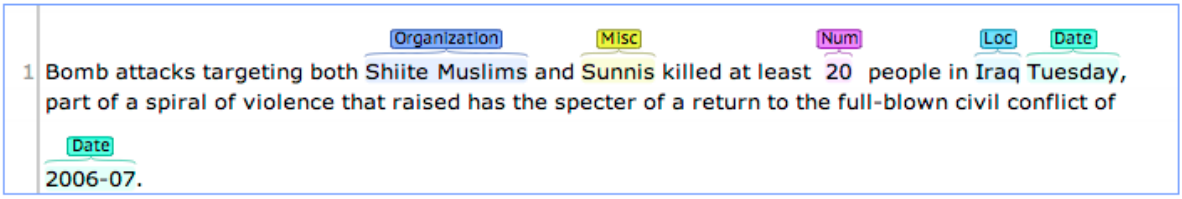
```
<EventID date="19950103" id="DEMO-04" category="DEMO">
<!-- [Paired events: LEFT_ generates a "visit" and "receive visit" events] -->
<EventCoding sourcecode="DAG" targetcode="GON" eventcode="032">
<EventCoding sourcecode="GON" targetcode="DAG" eventcode="033">
Dagolath's first Deputy Prime Minister Telemar left for
Minas Tirith on Wednesday for meetings of the joint transport
committee with Arnor, the Dagolathi news agency reported.
(ROOT
  (S
    (S
      (NP
        (NP (NNP Dagolath) (POS 's))
        (ADJP (JJ first))
        (NNP Deputy) (NNP Prime) (NNP Minister) (NNP Telemar))
      (VP (VBD left)
        (PP (IN for)
          (NP
            (NP (NNP Minas) (NNP Tirith))
            (PP (IN on)
              (NP (NNP Wednesday)))))
        (PP (IN for)
          (NP
            (NP (NNS meetings))
            (PP (IN of)
              (NP
                (NP (DT the) (JJ joint) (NN transport) (NN committee))
                (PP (IN with)
                  (NP (NNP Arnor)))))))))
    (, ,)
    (NP (DT the) (NNP Dagolathi) (NN news) (NN agency))
    (VP (VBD reported))
    (. .)))
```

Figure 1: PennTreebank example in a PETRARCH unit-test input frame

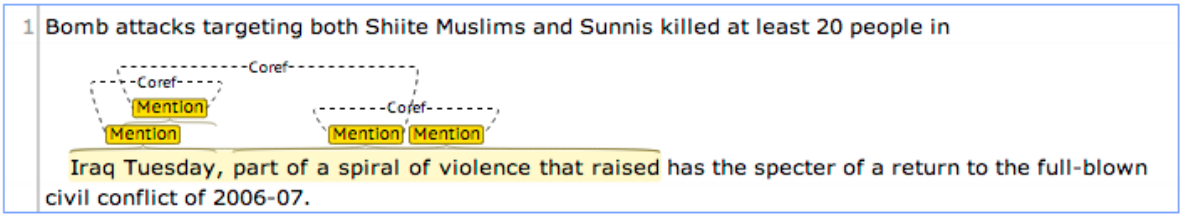
Part-of-Speech:



Named Entity Recognition:



Coreference:



Basic dependencies:

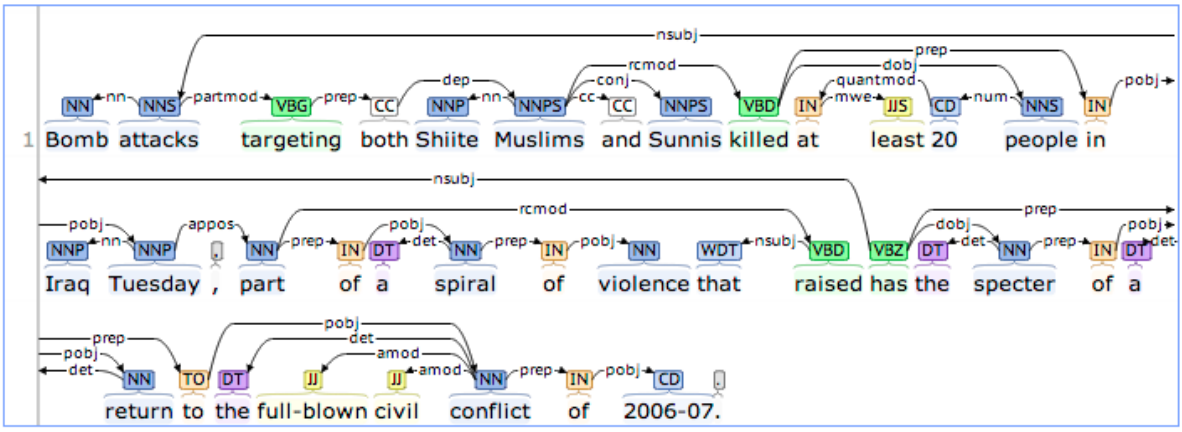


Figure 2: Additional markup and analysis provided by the Stanford CoreNLP system

1.6 Deduplication

Historically, deduplication has been an important part of the event data generation process when automated coding is used. It was originally used because of multiple reports of the same event—for example a meeting or a terror attack can easily generate tens of reports dealing with the same occurrence—and become even more important when multiple local sources are used, as is the case with news aggregators, since these will frequently reprint wire service stories, sometimes with editing, sometimes not.

In the current environment of web-based sourcing, deduplication turns out to be a critical aspects, and if done sloppily, can actually result in very significant *proliferation* of events, very significantly increasing the level of noise in the dataset. However, the number of duplicates—if these are accurately characterized—and the media outlets where those duplicates occur may be useful as an indicator of how important a story is, at least to the media. Consequently the ELDI system keeps track of all of the URLs which were determined to be duplicate, and provides these in a file of records linked to the event records. This provides additional transparency and the possibility of checking on the accuracy of the deduplication.

1.7 GeoData

In order to geolocate the coded events, ELDI makes use of the GeoText project developed by the GeoVista Center at Pennsylvania State University. Many other geocoders exist such as CLAVIN (`http://clavin.bericotechnologies.com/`), and geolocation is still not a solved problem in the relevant fields. Given this, the GeoText project is still an area of active research for the GeoVista center. More information, and documentation, can be found at `http://www.geotxt.org/` and `http://www.geotxt.org/api/`. Even more difficult than identifying location words within a given set of text is the accurate location of a single event. A text may mention several different locations, e.g., the President speaking from the Rose Garden, attackers from Aleppo, etc., but it is necessary to identify *one* as the single location for the event. As with the broader issue of geocoding, this is not a solved problem and will likely require a fair amount of active research to solve.

1.8 Custom feature detectors

Discussions and queries in several venues over the past year indicate that the greatest unfilled need in event-related coding deals with the detail coding of large-scale protest: this would

involve not only the simple event and location coding of the protest itself, but also features such as the number of people attending, the topics being addressed by the protest, and the response of the authorities. Reports of protests also generally have specific components that could be extracted more efficiently with specialized routines than with a generic event coder. We have secured NSF funding for an interdisciplinary effort to develop customized software for this task and will incorporate it.

1.9 LDC *Gigaword* corpus coding

While the ELDI system has been designed for real-time coding, we intend to code a reasonably long—and critically, totally transparent and free of IP issues—set of data using the Linguistic Data Consortium *Gigaword* corpus [Parker et al., 2011] of major news services that was developed under the DARPA GALE project, covers 2000-2010 and can be easily licensed. The Gigaword corpus has conveniently already been parsed using SCNLP so we will not need to do that pre-processing, though simply coding it will be a fairly large task, though we believe that we can get the required computing resources in collaboration with UT/Dallas.

2 PETRARCH

PETRARCH [Python Engine for Text Resolution And Related Coding Hierarchy[8]; PETR] is the Python-based successor to Tabari. The key differences are

2.1 Computer language

While thoroughly debugged, the code base of Tabari is about twelve years old and written in the computer language C++, which no longer has a large (or young) programming community. Python is *much* more suited than C++ for text processing and has a very broad and sophisticated user community focused on practical applications, for example with *scikit-learn* machine learning package, the *nltk* natural language processing package and other routines.

Python provides a number of additional advantages

- Open source (of course...tools want to be free...)

[8]Okay, so that works as a not unreasonable backronym, but for the most part it is "PETRARCH" because Schrodt think the guy was cool...

- Standardized across platforms and widely available/documented
- Automatic memory management (unlike C/C++)
- Generally more coherent than perl, particularly when dealing with large programs
- Text oriented rather than GUI oriented (unlike Java)
- Extensive libraries but these are optional (unlike Java)
- It may be possible to integrate C code at critical points for high-performance applications

The one clear advantage of TABARI over PETR is speed: TABARI coded around 2000 sentences per second. The current version of PETR, in contrast, codes only about 150 Treebank-parsed sentences per second, and SCNLP is *quite* slow, parsing only about 5 sentences per second in our tests. Given the availability of both stand-alone and cloud-based cluster computers, this speed is probably sufficient, though the SCNLP will require very substantial resources for any large set of texts (though only has to be done once, even if the coding dictionaries change).

2.2 Fully-parsed input

TABARI was a dictionary-based shallow parser. While multiple independent tests demonstrated this approach was sufficient to produce research-quality data, full parsing provides a number of advantages:

- Shallow parsing resulted in a number of cases where source and target were reversed. In *statistical* applications, this had little or no impact: the dominant feature of all event data sets is reciprocity, and correct identification of the relevant actors explains at least half of the variance in many applications—but is upsetting to naïve users.
- Shallow parsing is heavily dependent on having the appropriate phrases available in dictionaries—in fact this is way the approach works at all—and consequently is more likely to make mistakes when dealign with new sources. TABARI, furthermore, was prone to erroneous identifier
- SCNLP provides very extensive noun/verb/adjective disambiguation: many words in English can be used in all three modes:
  - "A protest occurred on Sunday" [noun]

  - "Demonstrators protested" [verb]
  - "Marchers carried protest signs" [adjective]

  TABARI , in contrast, relied on its relative limited dictionaries for disambiguation, though a large number of null-coded phrases—that is, phrases which simply keep sequences of words from being coded, rather than generating events—were in the dictionaries simply to handle disambiguation. The SCNLP input should be more robust, and will probably eliminate the need for a number of the null-coded dictionary entries.

- The use of parsed input allow for the identification of all named entities through noun phrases, whereas TABARI required all actors that might be coded as a source or target of an event to be in existing dictionaries. PETR will always pull these out whenever they occur in the source or target position and the resulting unidentified cases can be separately processed with named-entity-resolution (NER) software
- We should eventually be able to use other NLP tools to get more sophisticated co-referencing of pronouns and other entity references, particularly across sentences, though this is a fairly difficult problem with imperfect solutions even at the cutting edges of NLP.

While full parsing will not eliminate all of the coding issues, nor will it eliminate the need for dictionaries, it should substantially reduce them—as well as reducing the overall false-positive rate, particularly with respect in the actors

The possible downside is that the existing verb phrase dictionaries may require substantial additional work: at this point in time, we still don't have a clear sense of this. Initial experiments point to the fully-parsed representations being considerably more brittle than the shallow parsed—as apparently trivial a change as substituting "that" for "the" may significantly change a parse, and hence PETR is less likely to get the right answer for the wrong reason.

2.3 WordNet-based dictionaries

Under earlier NSF funding, we produced new versions of the verb dictionaries that are organized around WordNet synonym sets for both the core verbs and common noun sets such as those using units of currency and types of weapons. These should make the dictionaries more robust, since a general phrase such as "Nation X has agreed to provide Nation Y *currency*-million in development aid" will need to be entered only once, rather than sepa-

rately for each currency. WordNet has also been used to produce a new "agents" dictionary for common nouns such as "police", "soldiers", "president" which is substantially larger and more systematic than the dictionary used in TABARI coding. We had originally intended to modify TABARI to work with these but that programming was never completed, and has been deferred to PETR.

2.4 Extensions

While TABARI was open-source, the code was not particularly friendly, and only a few new features were added only the program had been released. In contrast PETR has been designed with numerous coding "hooks" for slotting in feature detectors: we currently have implemented a very extensive "issues" list which allows Boolean exclusion criteria and can work at either the sentence or story level.

2.5 Parallel deployment

Given the size and number of text documents that any event-data project must code, all of these projects are developed with an eye towards easy parallel deployment. In other words, our aim from the beginning has been to develop an ecosystem that can easily handle large corpora of text in a modern computing environment. The most basic form of parallel processing focuses on splitting the raw, underlying data across the appropriate computational units, e.g., CPUs or nodes in a cluster. While the PETR event data coder is slower than the previous-generation TABARI coder, a more significant bottleneck comes from the StanfordNLP processing stage. The problem of parallelizing StanfordNLP is itself not an easy task.[9] This step is made even more difficult by the need to interoperate Java code (StanfordNLP) and a Python codebase (PETRARCH). While we have not managed to solve this problem in a satisfactory manner yet, easy parallel deployment of the entire pipeline is a high priority for the project, and once the code base has stablized, we expect to release it completely configured to run on one or more of the commercially-available "cloud" cluster computing services. The web scraper described above does, however, currently run in parallel.[10]

[9] See #23 in the FAQ for StanfordNLP at `http://nlp.stanford.edu/software/parser-faq.shtml`.
[10] Turns out that scraping 166 RSS feeds takes more than an hour when run in serial.

3 Open Event Data Alliance [OEDA]

Figure 3: Jack, the OEDA mascot `http://openeventdata.org/jack_info.html`

Because of the recognition that event data has reached a point where, with relatively little up-front investment in learning how to use available open source tools, data can now be produced at zero marginal cost. However, this has also led to a series of unfortunately events and some unpleasant experiences—hardly atypical for open source projects in their early stages but still less than optimal—and there is a clear need to provide a greater level of institutionalization to the field. In response to this, we are working with a number of groups to develop The Open Event Data Alliance (OEDA) is a consortium, in the process of incorporation as a limited liability corporation,[11] of for-profit organizations, not-for-profit organizations, and individuals committed to facilitating the development, adoption and analysis of social and political event data through

- the provision of open access datasets
- the development of open source software and coding ontologies for these purposes, and
- the creation of standards for documenting the source data used for event data as texts cannot be typically directly shared because of licensing and intellectual property concerns.

The OEDA does not seek to establish any definitive imprimatur but rather to provide guidance for voluntary solutions to coordination problems on issues and resources of common concerns.

The prime objective of the OEDA is to provide reliable, open access, multi-sourced political event datasets that are updated on at least a daily to weekly basis, which are transparent

[11] We are already receiving junk mail offering to print us custom checks and develop our web site, so this must be real.

and documented with respect to the origin of the source texts, and use one or more of the open coding ontologies supported by the organization. We hope to be able to do this solely as an aggregator rather than generating such data—in particular we expect to be linking to multiple data sets, either sharing a common format or supported by software that will translate into that format—but the latter will be kept open as an option if this is necessary to insure a reliable resource. When suitable sources are available, preference will be given to data which are also transparent with respect to the use of open source coding engines and dictionaries, but the organization will support data sets which have been produced in part or in total using proprietary methods, provided the resulting data are open access, documented, and clear of intellectual property issues.

The objective of the OEDA is not to provide "one data set to rule them all" and as such it takes no position on the relative appropriateness or validity of various event datasets, although it may support research on scientific comparisons of various data sets, including those provided by its members. Rather we seek to provide a stable, credible and recognized data source that can be used to support the research and development efforts of the community.

Following the model of the Linguistic Data Consortium, as opportunities arise, the OEDA expects to negotiate and become the intellectual property license holder for news texts that can be used to produce open access event data sets. These activities could include licenses that restrict the text (but not the derived data) to use within the organization, licenses that provide access only to the membership, and licenses that provide for open access.

The organization will not seek to compete with its membership for the provision of research and analysis. It will undertake the development of open source software and standards in circumstances where these needs have been identified that are not being met in a timely fashion as determined by its boards of directors and board of advisors. When opportunities arise, the organization will actively seek funding for workshops on common standards and for training.

As resources and voluntary efforts allow, the OEDA expects to participate in the conferences and workshops of other professional organizations, both to disseminate results of its own work and to provide training, While the organization may on occasion initiate independent conferences, these are not the primary purpose of the organization. We do not anticipate the establishment of a journal, magazine or any other periodic communication, but will maintain various forms of electronic communications. The organization will give no awards. Particularly those involving the recruitment of committees.

In summary, we anticipate the OEDA engaging in the following activities

- Maintenance of a set of reliable—24/7/365 availability—internet-based open access, multi-sourced social and political event data sets that are updated on at least a daily or weekly basis. When practical, this will be done by aggregating existing data sets but if needed the organization will generate its own data using open source resources. This data will be maintained in reliable, mirrored archives.
- Licensing of texts of news sources that can be used for the production of open access data.
- Development of specialized coding software, particularly computationally-intensive tasks such as parsing and machine translation
- Dictionaries and quality control
  - Named-entity updates
  - Expansion/refinement of the actor, event and thematic ontologies
  - Maintaining dictionary development expertise
  - Managing the core open source software
- Maintenance of a blog, listserv and servers, as well as contributing in other electronic media as may arise through future social and technological developments. The servers will contain a versioned archive of various open-source resources.
- Development of voluntary open standards for resources such as data formats (and software to translate between formats), coding ontologies, and dictionaries, and development of "best practices" for coding protocols.
- As resources permit, legal defense of any challenges to open source materials, or coordination with other organizations such as the Electronic Frontier Foundation or the Bill of Rights Defense Committee on these efforts.
- Sponsorship of training and best-practice workshops at professional meetings as opportunities allow. If there is sufficient demand, sponsorship of one or more independent conferences

4 Open Issues

At the present time, we would identify two major open issues that we feel need to be resolved—in an open and transparent environment—

4.1 The extension existing event and actor ontologies

At present, CAMEO [Schrodt, 2012a, Schrodt et al., 2009] and IDEA [Bond et al., 2003] are the only two event coding ontologies in wide use for event data, though a variety of actor-level coding systems exist in various conflict data sets [Bernauer and Gleditsch, 2012, Schrodt, 2012b]. The advantage of CAMEO is that it modified the older systems to be more appropriate for automated coding, it is thoroughly documented, it has been implemented in open-source dictionaries which have been used in multiple projects and it contains a very extensive ontology for coding actors; the disadvantage is that it was developed specifically to code international mediation and has not been extended to deal with events outside of the traditional political interactions. The advantage of IDEA is that it contains a number of extensions and integrated the half dozen or so different ontologies available in the early 2000s. The disadvantage is that it retains a number of legacy codings from the pre-automated coding era, the documentation is very sketchy, it has no actor ontology comparable to that found in CAMEO, and there is only a single proprietary implementation.

Schrodt and Bagozzi [2013] have demonstrated that the existing event ontologies are probably picking up only about half of the events that are arguably "political", with the single biggest missing category being routine democratic processes (elections and parliamentary debate). More generally, the core extensions probably need to be

- natural disaster
- disease
- criminal activity
- financial activity
- refugees and related humanitarian issues
- human rights violations
- electoral and parliamentary activity

The growing widespread interest in event data beyond the study of conflict highlights the need to for more specialized coding schemes...indeed, the development and maintenance of specialized ontologies is one primary objective of the Open Event Data Initiative.

While both CAMEO and IDEA have been used as comprehensive schemes—and IDEA was explicitly designed with this in mind—we are not entirely convinced this is practical as the event data world expands, since different projects require more or less detail, and mainte-

nance of a single scheme that entails all possible distinctions which might be relevant to political analysis could quickly become impractical and, being impractical, won't be used. The alternative is to encourage the development of multiple coding frameworks that are compatible at some level—perhaps the 2- and 3-character codes—but not all levels. Similar efforts could be devoted to actor coding: there are about ten coding systems in wide use—including multiple ISO systems, multiple COW systems, multiple FIPS systems—for national-state actors, but sub-state actors are far less standardized. Efforts are underway to address this and should be addressed, though any such efforts are labor-intensive.

4.2 Automating the development and updating of dictionaries

4.2.1 Actors

The TABARI system has very extensive open-source dictionaries for the identification of political actors. Central to these is the 32,000-line `CountryInfo.txt`, a general purpose file intended to facilitate natural language processing of news reports and political texts. This covers about 240 countries and administrative units (e.g. American Samoa, Christmas Island, Hong Kong, Greenland); fields include adjectival forms and synonyms of the country name, the capital city and cities with populations over 1-million, regions and geographical features (*WordNet*meronyms), leaders from `http://rulers.org` and members of government from the *CIA World Leaders* open database. This has been supplemented by names obtained from lists of major corporations, NGOs and IGOs, and some militarized groups such as al-Qaeda.

`CountryInfo.txt` and the ancillary dictionaries do not, however, catch all names: For example, it does not have opposition leaders unless these have been in government, nor the names of new militarized groups, nor does it provide all equivalent forms of a names, e.g. "President Obama," "President Barak Obama," "United States President Barak Hussein Obama" and so forth. To some extent, this process of identifying equivalent forms can be automated, and there are some very sophisticated NER methods available in open source software–for example using conditional random fields and hidden Markov models–though the NER issue is relatively simple for political actors found in news reports, who usually have regularized names and titles.

With fully-parsed text, NER is fairly straightforward—actors are always noun phrases—and can be done this with modest enhancements of the existing systems, but it hasn't been done. More generally, there is a very large literature on NER that hasn't been fully incorporated into the existing systems. Most of the work, in fact, is not actually the identification of new

actors—once an existing actor list has been established, political significant actors persist for years and often decades—but disambiguating actors via the phrases that correspond to these. New actors are generally introduced in context, so it is relatively easy to figure out the appropriate codes. Network-based approaches such as the combination of Bayesian, network and rule-based approaches instantiated in Getoor's entity resolution system [Getoor and Machanavajjhala, 2013] are very promising in this regard.[12]

That said, it is easy to exaggerate the level of effort required here: as shown in Tables 1 and 2 references to political actors are highly asymmetrically distributed with an extremely long tail—essentially a classical rank-size distribution—and a small amount of effort devoted to those which are frequently mentioned will be sufficient to get almost all of the useful information. While practical automated methods should be encouraged, there is little reason to invest millions in specialized software (which may in the end fail anyway) for a task that could be more effectively done by a small number of trusted and trained coders working a few tens of hours per week at $20/hour.

Table 1: Actor distribution: high frequency cases. Source texts from the research phase of the DARPA ICEWS project [O'Brien, 2010] and consequently emphasize Asian actors

Phrase	Proportion of events
China	11.86%
United States	11.56%
Russian Federation	8.43%
Japan	7.99%
North Korea	5.33%
India	5.24%
South Korea	3.45%
Chinese	3.44%
UN	3.14%
Taiwan	3.13%
Pakistan	3.10%
Thailand	2.88%
Australia	2.48%
Iraq	2.23%
United Kingdom	2.08%
Indonesia	1.96%

[12]See also `http://www.youtube.com/watch?v=Vo-v3ptPmdQ`

Table 2: Actor distribution: sample of the tail from the research phase of the DARPA ICEWS O'Brien [2010] project

Phrase	Proportion of events
Hamid Karzai	0.01%
President (Angola)	0.01%
Yang Hyong Sop	0.01%
Kashmir State	0.01%
Ehud Olmert	0.01%
Police (Sri Lanka)	0.01%
Vojislav Kostunica	0.01%
Commerce Minist (India)	0.01%
Parliament (Iran)	0.01%
President (Yemen)	0.01%
Foreign Minist (Netherlands)	0.01%
Director General (IAEA)	0.01%
Liu Qi	0.01%
Yang Jiechi	0.01%
Business (Hong Kong)	0.01%
President (Namibia)	0.01%
Police (China)	0.01%
Business (France)	0.01%

4.2.2 Events

While this problem is not trivial, with fully-parsed texts it is fairly straightforward: one simply looks for sentences containing actors as subjects and objects that are not coding with the existing verb phrases, cluster these sentences based on word counts with synonym sets, then identify the distinguishing phrases. That said, there are actually related four issues here:

- Incorporating an entirely new class of events into an existing ontology, essentially an orthogonality issue;
- Filling in the details of an existing class that has been properly classified at the cue category level;
- Dealing with misclassification of events which should be coded; and
- Dealing with false positives that shouldn't be coded at all, which as noted earlier is probably a standard text classification task for which mature technologies exist.

We also believe more effort should be made on establishing an open and replicable framework for event dictionary development. Such a framework would be more fruitful for the

continued diffusion of event data in other specialized domains and should leverage existing automated dictionary development algorithms and techniques [e.g., Riloff, 1996] for event pattern extraction. These algorithms take as inputs collections of relevant (in-domain) and irrelevant (out-of-domain) texts, event phrases, and event characteristics for learning and can be modified to extend existing event dictionaries or develop new specialized ones.[13]

5 Conclusion

Rifkin [2014] recently observed that the most radical transition of the past decade is technologies which combine network effects with zero marginal production costs. While once ridiculed as the domain of naïve hippies with questionable personal hygiene, open source has come to completely dominate the world of computer technology, including core applications such as operating systems, web servers, all major programming languages, and version control systems.[14] However, the now-extensive literature on open source shows that successful projects do not operate in some Randian utopia of anarchic self-interest but function more along the lines of a guild—or perhaps a baboon troop—with a core coordinating group, mutually accepted open standards, and extensive collaboration and coordination with other organizations.

Our hope is that EL:DIABLO, PETRARCH and the institutional framework of the OEDA will contribute to replicating the earlier success of the programming (gcc, Linux, Python) and statistical (*R*) communities in creating a self-reinforcing network of trust, collaboration and the sharing of resources for analysis and tool development. In the contemporary environment, event data is simply too big an opportunity *not* to open source. These professional communities have demonstrated that once a common platform has been established that allows information to be shared with an guarantee that it will remain accessible, open source approaches completely outpace proprietary approaches in quality, availability, speed and flexibility.

[13]See research by NLP Research Group at University of Utah: `http://www.cs.utah.edu/ riloff/publications.html`

[14]Schrodt, having escaped the bonds of institutional software licensing and—critically—clueless IT administrators, currently works on machines with only two programs that are not open source: the Macintosh operating system and the BBEdit text editor.

References

Edward E. Azar. The conflict and peace data bank (COPDAB) project. *Journal of Conflict Resolution*, 24:143–152, 1980.

Thomas Bernauer and Nils Petter Gleditsch. Special issue: Event data. *International Interactions*, 38(4), August 2012.

Doug Bond, Joe Bond, Churl Oh, J. Craig Jenkins, and Charles L. Taylor. Integrated data for events analysis (IDEA): An event typology for automated events data development. *Journal of Peace Research*, 40(6):733–745, 2003.

Deborah J. Gerner, Philip A. Schrodt, Ronald A. Francisco, and Judith L. Weddle. The machine coding of events from regional and international sources. *International Studies Quarterly*, 38:91–119, 1994.

Lise Getoor and Ashwin Machanavajjhala. Entity resolution for big data. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1527–1527, New York, NY, USA, 2013. ACM. URL `http://doi.acm.org/10.1145/2487575.2506179`.

Gary King and Will Lowe. An automated information extraction tool for international conflict data with performance as good as human coders: A rare events evaluation design. *International Organization*, 57(3):617–642, 2004.

Russell J Leng. *Behavioral Correlates of War, 1816-1975. (ICPSR 8606)*. Inter-University Consortium for Political and Social Research, Ann Arbor, 1987.

Charles A. McClelland. *World Event/Interaction Survey Codebook (ICPSR 5211)*. Inter-University Consortium for Political and Social Research, Ann Arbor, 1976.

Sean P. O'Brien. Crisis early warning and decision support: Contemporary approaches and thoughts on future research. *International Studies Review*, 12(1):87–104, 2010.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. *English Gigaword Fifth Edition*. Linguistic Data Consortium, Philadelphia, 2011.

Jeremy Rifkin. The rise of anti-capitalism, March 2014. URL `http://www.nytimes.com/2014/03/16/opinion/sunday/the-rise-of-anti-capitalism.html`.

Philip A. Schrodt. Twenty years of the Kansas event data system project. *The Political Methodologist*, 14(1):2–8, 2006.

Philip A. Schrodt. *Conflict and Mediation Event Observations (CAMEO) Codebook*. `http://eventdata.psu.edu/data.dir/cameo.html`, 2012a.

Philip A. Schrodt. Precedents, progress and prospects in political event data. *International Interactions*, 38(4):546–569, August 2012b.

Philip A. Schrodt. Automating the extraction of political indicators from text sources on the web: Event data. Presented at the International Studies Association, Toronto, March 2014., 2014. URL `http://eventdata.parusanalytics.com/papers.dir/Schrodt-ISA14.pdf`.

Philip A. Schrodt and Benjamin Bagozzi. Detecting the dimensions of news reports using latent dirichlet allocation models. International Studies Association, April 2013. URL `http://eventdata.parusanalytics.com/papers.dir/BagozziSchrodt.EPSA12.pdf`.

Philip A. Schrodt and Deborah J. Gerner. Validity assessment of a machine-coded event data set for the Middle East, 1982-1992. *American Journal of Political Science*, 38:825–854, 1994.

Philip A. Schrodt, Deborah J. Gerner, and Ömür Yilmaz. Conflict and mediation event observations (CAMEO): An event data framework for a post Cold War world. In Jacob Bercovitch and Scott Gartner, editors, *International Conflict Mediation: New Approaches and Findings*. Routledge, New York, 2009.