Automated Coding of Political Event Data

Philip A. Schrodt and David Van Brackle

1 Introduction and Overview

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 [1] and Charles McClelland's WEIS [18] as well as a variety of more specialized efforts such as Leng's BCOW [16]. By the late 1980s, the NSF-funded *Data Development in International Relations* project [20] had identified event data as the second most common form of data—behind the various Correlates of War data sets—used in quantitative studies. The 1990s saw the development of two practical automated event data coding systems, the NSF-funded KEDS (http://eventdata.psu.edu; [9, 31, 33]) and the proprietary VRA-Reader (http://vranet.com; [15, 27]) and in the 2000s, the development of two new political event coding ontologies—CAMEO [34] and IDEA [4, 27]—designed for implementation in automated coding systems. A summary of the current status of political event projects, as well as detailed discussions of some of these, can be found in [10, 32].

While these efforts had built a substantial foundation for event data—by the mid-2000s, virtually all refereed articles in political science journal used machine-coded, rather than human-coded, event data—the overall development of new technology remained relatively small. This situation changed with the DARPA-funded Integrated Conflict Early Warning System (ICEWS; [25, 26]) which utilized event data development coded with automated methods. The key difference between

P.A. Schrodt (✉)
Political Science, Pennsylvania State University, University Park, PA 16801, USA
e-mail: schrodt@psu.edu

D. Van Brackle
Lockheed Martin Advanced Technology Laboratories, Lockheed Martin Advanced Technology Laboratories 3550 George Busbee Parkway, Kennesaw, GA 30144, USA
e-mail: dvanbrac@atl.lmco.com

V.S. Subrahmanian (ed.), *Handbook of Computational Approaches to Counterterrorism*,
DOI 10.1007/978-1-4614-5311-6_2,
© Springer Science+Business Media New York 2013

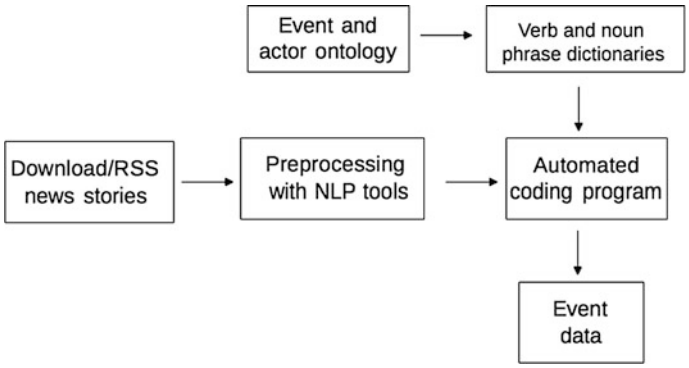

Fig. 1 Process of generating event data by automated methods

the ICEWS event data coding efforts and those of earlier NSF-funded efforts was the scale. As O'Brien—the ICEWS project director—notes,

> . . . the ICEWS performers used input data from a variety of sources. Notably, they collected 6.5 million news stories about countries in the Pacific Command (PACOM) AOR [area of responsibility] for the period 1998-2006. This resulted in a dataset about two orders of magnitude greater than any other with which we are aware. These stories comprise 253 million lines of text and came from over 75 international sources (AP, UPI, and BBC Monitor) as well as regional sources (*India Today, Jakarta Post, Pakistan Newswire,* and *Saigon Times*).

The later phases of ICEWS [25] moved to near-real-time global event data production and the scale of this coding effort increased even further, covering 175 countries and nearly 20-million stories [36].

This chapter will describe a number of incremental improvements and lessons-learned in the recent experience of both our open-source work at Kansas and Penn State, which supported both ICEWS and National Science Foundation-funded basic research, and the proprietary work of the Lockheed Martin Advanced Technology Laboratories, which made several important extensions to that work in conjunction with ICEWS. This chapter is a "how-to" exercise—albeit at a rather high level of generality in places—rather than a theoretical one, and the objective is to provide some guideposts for others who might be interested in undertaking similar efforts, whether as basic research or for applied policy purposes. The chapter essentially goes through the various phases of a machine-coding project, outlined schematically in Fig. 1, starting with the decision on whether to use human coding at all, and discusses both the issues we encountered, the choices we made for resolving these, and thoughts on further developments that might be relevant in the future.

From the outset, we would emphasize that automated coding is a work in progress. It has clearly crossed the threshold into the realm of practical utility—ICEWS models which use event data perform much better than human forecasters—but we do not view it as fully developed. In addition, we are making increasing use of pre-processing software from the much larger field of computational natural

language processing, and advances in that area will undoubtedly substantially increase the accuracy of our methods, and quite possibly open avenues for additional coding in areas such as geolocating events, sentiment analysis, coding texts in languages other than English, and resolution of long-standing NLP issues such as noun-verb disambiguation in English, and pronoun co-referencing. Finally, this discussion deals with the field from the perspective of a specific line of related coding programs—KEDS, TABARI and JABARI-NLP—and some of these issues will differ for coding systems using alternative approaches.

1.1 Human Versus Machine Coding

As discussed in some detail in [25], in some circles, automated coding and statistical forecasting can be a very hard sell: many people simply cannot believe that a purely statistical model, generated with well-understood formal methods that are 100 % transparent, and using data generating by automated coding techniques that are also 100 % transparent, can do better than their anything-but-transparent intuition. This is not a problem unique to event data analysis: Nobel Prize-winning psychologist Daniel Kahneman [14, Part III, "Overconfidence"] provides numerous examples from a diverse set of behavioral domains where humans believe they can outperform statistical methods (or dart-throwing chimpanzees) despite overwhelming evidence to the contrary.

Still, before embarking on a coding exercise, you will probably first need to convince skeptical humans. Who will not be impressed by comparisons to chimpanzees, and who usually demonstrate the inferiority of automated methods by pointing to an incorrectly coded sentence—and any event data system, human or machine, will have plenty of those. Meanwhile ignoring the fact that the *total* amount of information in the system is vastly greater than that which can be processed by an individual, and while the intuitive analysis may be better in an individual case (and certainly for an individual news report), the *composite* has better performance. A subject-matter-expert (SME) may perform better on their area of expertise in a particular time frame (though Tetlock's research [37] would suggest not even this is true), but there is little evidence that they can perform broadly. In contrast, using event data, the ICEWS forecasting models predicted five indicators for 29 countries at a monthly granularity for almost 15 years, and effort this is now being scaled to cover the entire world.

As noted in [30], if one is using event data in forecasting models—the objective of ICEWS and most other applications of event data—coding error is only one potential source of error that lies between "events on the ground" and the predictions of the forecasting model. These include

- News reports are only a tiny, tiny fraction of all of the events that occur daily, and are non-randomly selected by reporters and editors;

- Event ontologies such as WEIS, CAMEO and IDEA are very generic and bin together events that may not always belong together in all contexts;
- Forecasting models always contain specification error and cannot consider everything; for example few if any political forecasting models contain a full economic forecasting component;
- Political systems have a degree of intrinsic randomness due to their inherent complexity, chaotic factors even in the deterministic components of those systems, the impact of effectively random natural phenomena such as earthquakes and weather, and finally the effects of free will, so the error intrinsic to a forecasting model will never reduce to zero.

In this chain of events, the impact of coding error in automated systems, while still relevant, is not necessarily dominant. The first and fourth factors also affect SME evaluations; the second and third affect statistical models based on human coding. And the bottom line is that in gold-standard, out-of-sample predictive tests, models using event data consistently show a higher level of predictive accuracy than is typical of SMEs subjected to systematic tests.

When assessing the alternative of human coding for generating event data, there are two additional problems. The first is simple impossibility. In the early phases of the ICEWS project, TABARI repeatedly coded 26-million records in 6 min, resulting in about 3-million events. Sustained human coding projects, once one takes in the issues of training, retraining, replacement, cross-coding, re-coding due to effects of coding drift and/or slacker-coders and so forth, usually ends up coding about six events per hour.[1] The arithmetic is obvious: 6 min of automated coding, or 500,000 labor-hours of manual coding, probably costing on the order of $10-million when labor and administrative costs are taken into effect. And for the manual coding, that amount will code the texts once.

For this reason, human-machine comparisons are of little practical consequence, since human coding is not an option. Multiple published tests [15, 33] have shown that machine coding is comparable in accuracy to human coding. But the human coding accuracy in some of those tests is quite low: King and Lowe [15] use an assortment of measures (and a fairly specific sampling method) but the accuracy on the individual VRA codes alone (Table 2, pg 631)—not the complete record with source and target identification, another major potential source of error—is in the range 25 % (!) to 50% for the detailed codes and 55–70 % for the cue categories. Similarly, [21] show that the reliability of the human coding in the widely-used Comparative Manifestos Project is less than half what is commonly reported, and for some indicators drops as low as 25 %; [28] show similar problems in the coding of governance events in UN peacekeeping. Human coding is anything but flawless.

[1] Individual coders, particularly working for short periods of time, can of course reliably code much faster than this. But for the *overall* labor requirements—that is, the total time invested in the enterprise divided by the resulting useable events—the six events per hour is a pretty good rule of thumb and—like the labor requirements of a string quartet—has changed little over time.

On a supplementary web site (http://eventdata.psu.edu/papers.dir/automated.html) Schrodt has provided an extended rebuttal of the claims in [6] for abysmally low coding accuracy for TABARI. Briefly, while [6] provide almost no information on what combination of software they actually tested, it appears that they attempted to evaluate the system using inappropriate dictionary files. Under any circumstances, it is simply impossible to reconcile their results with the independent assessment of ICEWS Phase I [26] which used the supposedly highly inaccurate data produced by TABARI and yet surpassed the ICEWS 80 % accuracy levels at the same time two competing projects using alternative sets of event data failed to meet those criteria.

Lockheed's internal assessments of the accuracy of TABARI on the initial Asian data evaluated in ICEWS Phase I was around 58 % [36]. This is likely lower than the TABARI accuracy in the Levant and Balkans data sets produced by NSF research—probably closer to 70 %—because during the ICEWS Phase I work relatively few changes were made to the verb-phrase dictionaries, which had been developed on those two regions (see Sect. 3.1). Subsequent work on the JABARI-NLP system during the second two phases of ICEWS brought the accuracy first to 71 % by the incorporation of open-source parsing into JABARI-NLP. Additional enhancement to dictionaries and the processing of various contingencies such as agents and the coding of actions without a clear target led to the current (October 2011) level of "an overall precision of 75.42% with a 3.10% confidence interval." [36]

As noted in greater detail in the web supplement, we don't have a contemporary large, randomly sampled human coded comparison data set—given the futility of human coding as an alternative to automated coding, no one has invested the very substantial amounts of time and money that would be required to do this. The *major* problem with such an exercise is reaching convergence among the human coders: about 10 years ago VRA undertook a substantial, well-designed exercise to do this but no results ever came of it, apparently because the coding never came close to a consensus. Based on our experience and anecdotal reports from various other event data coding projects (Maryland's GEDS, the CACI project for the NSC 1981–1985, Third Point Systems for the Saudis in the 1980s, Russ Leng's BCOW at Middlebury) over the years, that sustained accuracy will be in the range of 70 % at best. The human-coded COPDAB data set somehow manages to miss the Korean War [12], the human-coded GEDS project, which consumed the bulk of the event data expenditures of the NSF-funded "Data Development in International Relations" project has not been used in a single refereed article.

This is not to say that continued efforts should not be made to improve the quality of event coding, and Table 1 provides some general guidelines for situations where human coding is preferable to automated coding. Furthermore, event data provides a "best possible case" for automated coding, since it extracts relatively simple information that usually corresponds to the basic subject-verb-object structure of a typical English-language sentence that is describing an interaction.

Finally, automated coding tools—as well as some of the other NLP software described below—can be effectively used in *machine-assisted coding*. The Chenoweth and Dugan project [7, 8] has used TABARI as a sophisticated pre-filter for coding

Table 1 Tradeoffs between human and automated coding

Advantage to human coding	Advantage to machine coding
Small data sets	Large data sets
Data coded only one time at a single site	Data coded over a period of time or across institutions
No relevant dictionaries	Existing dictionaries can be modified
Complex sentence structure	Simple sentence structures
Metaphorical, idiomatic, or time-dependent text	Literal, present-tense text
Money available to fund coders and supervisors	Money is limited

incidents of terrorism, with a substantial reduction in the required labor costs, and the SPEED event data project [23, 24] uses a variety of customized NLP tools for this purpose.

2 Text Acquisition and Formatting

The first step in generating event data is the acquisition of news reports to code. Following the lead of most event data projects, we initially relied primarily on the Lexis-Nexis (LN) data service; in some of the initial phases of the project these were downloaded; in later phases they were acquired in bulk directly from LN by Lockheed, though this apparently involved the use of the same search engine that is available to ordinary users.

The two key differences between this project and most earlier event data projects was the sheer magnitude of the downloads, and the fact that we were using multiple sources. The eventual text corpus for 1997–2009—after initial filtering—involved about 30 GB of text, which reduced to about eight-million stories.[2] Second, unlike most earlier projects that used a small number of sources—typically the international newswires Agence France Press, BBC, Associated Press and United Press International—we used about 30 different regional sources.[3]

LN, unfortunately, proved problematic, as we also found in [35]. In all likelihood, this is due to LN using a legacy system that was designed to do very narrow searches, rather than providing a large-scale data dumps. In the later phases of the project, we switched to the newer Factiva service [36]. This does not appear to have these search engine problems, presumably because it is working with a relatively new system, and also provided stories from the Reuters news agency. Data providers

[2]The count of "stories" has varied continually as we've updated the downloads, modified the filters and so forth, and so an exact count is both unavailable and irrelevant. But starts around around eight to nine-million.

[3]We've actually identified about 75 distinct sources in the stories, presumably the result of quirks in the LN search engine. However, these additional sources generate only a small number of stories, and by far the bulk of the stories come from the sources we had deliberately identified.

appear to be gradually becoming accustomed to bulk requests that will be used for data-mining, and it is quite possible that these resources will become more available in the future.

The use of multiple sources provides a challenge in extracting the required information—the date, source and individual sentences—from the original download. Following the earlier work in the KEDS project, we were largely using source-specific filters, generally in `perl`. While LN and Factiva are generally consistently formatted, the diverse set of sources—and the sheer size of the files—proved a challenge, particularly since the local sources are more likely to contain minor quirks that will throw off a filter.

As we had discovered in earlier projects, in many sources the task of sentence delineation is a major challenge, both due to the presence of abbreviations, the occasional formatting errors that will cause sentences or entire paragraphs to run together, and the presence of a very large amount of non-sentence material such as tables of sports scores, exchange rates and commodity prices, chronologies, news summaries, weather reports and other such material. In principle, a suitably complex Boolean search term should exclude these; in practice one can't depend on this, particularly for the regional sources. These exceptions are sufficiently widely varied that it is nearly impossible to eliminate all of this using rules on the story itself—though we did have about 30 or so simple rules based on the headline of the story—and instead one needs to use more general rules such as the length of the "sentence." Most news sentences are around 150–300 characters in length, and anything below about 40 characters is almost certainly not codeable. There are also a few patterns easily written as regular expressions that will identify non-material: For example something of the form `\d+\-\d+` is almost always a sports score.

2.1 Filtering: Irrelevant Stories

Irrelevant stories have been the bane of the event data source texts from the beginning of our experience. For example, the search string for the now-30-year KEDS "Levant" data set primarily looks for stories containing the names or synonyms of the six actors tracked in the data set: Egypt, Israel, Jordan, Lebanon, the Palestinians, and Syria. However, our early downloads covered the peak of the career of basketball player Michael Jordan and we ended up with quite a number of basketball stories. These are relatively harmless and easily discarded by TABARI or Boolean search exclusions, but they do present problems when downloading—we originally did this using a phone modem [31]—or when one is paying by the story.

However, other types of stories are much more problematic. The most important are chronologies and retrospectives, which describe political events that occurred in the sometimes distant past, yet the dateline of the story is in the present. A good example would be various World War II commemorations, which typically receive extensive coverage and could be miscoded as conflict behavior between the US, Germany and Japan. Recent enhancements to JABARI-NLP specifically address these contingencies.

Another longstanding problem are international sports competitions that use military metaphors. World Cup reports, for example, always use the simple national names—Netherlands versus Spain—and not infrequently use terms such as "battle," "fought," "standoff" and the like. These can usually be solved by discard phrases—a TABARI discard phrase causes the story to be skipped if the phrase occurs anywhere in the text—involving every imaginable form of competition, sporting and others. But even this will fail when the sports context is implicit, such as a [hypothetical] report on the World Cup final on 11 July 2010 that might begin, with little concern that it will be misinterpreted, "Fans eagerly await tonight's battle between the Netherlands and Spain." Furthermore the sheer volume of such stories—as much as a third of the stories in areas where little seems to be happening except sports—decidedly increases download times and costs.

2.2 Filtering: Duplicates

The news downloads contain a very large number of stories that are either literally duplicates, or else are effectively duplicates. These generally come from five sources

- Exact duplicates, where a local source simply reprints the contents of an international newswire story. This is what newswires are for, so it happens a lot;
- Multiple reports of the same event—for example a suicide bombing—as it develops; AFP does this frequently;
- Stories repeated to correct minor errors such as incorrect dates or spelling;
- Lead sentences that occur in general news summaries—which may occur multiple times during a day—as well as in the story itself;
- Multiple independent reports of the event from different news sources: this was a major issue because of the large number of stories we were coding.

Duplicate detection is a very difficult problem, particularly when multiple sources are involved. We dealt with exact and near duplicates by simply seeing whether the first 48 characters of the story matched—this obviously will catch all duplicates and tends to catch minor duplicates such as corrections of spelling errors much of the time.[4] Cross-source duplicates are dealt with using the *One-A-Day* filter discussed below.

When used in a predictive mode, as we are doing with ICEWS, duplicates are not necessarily a bad thing, since they generally will amplify politically-relevant

[4]This will not, however, catching spelling corrections in the first 48 characters. In the Reuters-based filtering for the KEDS project, we did a count of the frequency of letters in the lead sentence, and identified a duplicate if the absolute distance between that vector for two stories, $\sum |x_i - y_i| > \eta$, where the threshold η was usually around 10. This catches spelling and date corrections, the most common source of duplicates in Reuters, but failed on AFP, which tends to expand the details in a sentence as more information becomes available.

signals. In other words, if reporters or editors think that something is important, it is more likely to be repeated, both within sources and across sources, than something that is mundane.

However, when trying to measure changes of "ground-truth" behavior against a baseline over a long period time, duplicates are a serious problem, both across sources and within sources. Cross-source duplication has probably changed considerably over the past 15 years due to local sources putting increasing amounts of material on the Web, and more generally the globalization of the news economy, so that events in once-obscure places are potentially of international interest.[5] In-source duplication can change due both to changes in the resources available to an organization—while not part of the ICEWS source set, Reuters went through something close to an organizational near-death experience during the period 1998–2002 [22] and the frequency of its reporting dropped dramatically during that time—and policies on updating, corrections and the broadcasting of summaries.

As discussed above, duplicate detection is a major challenge in the current environment. Improved story classification to identify, for example, sports stories, historical chronologies and movie reviews, also would simplify the dictionaries by eliminating the need for a number of discard and null-coded phrases that are present only to avoid coding stories that shouldn't be in the data stream in the first place.

Duplicate detection is a fairly specialized application, and one where we've yet to find much in the way of open source software. However, our sense is that algorithms considerably more sophisticated than those we are using exist in various proprietary aggregation systems, notably Google News, European Media Monitor (http://emm.newsbrief.eu/overview.html), and the non-open-source academic project NewsBlaster (http://newsblaster.cs.columbia.edu/). A more thorough review of the computer science literature might produce some guidance on these issues.

In addition, there is a rich literature with well-documented and robust methods—notably support vector machines—for document classification, and these may work considerably better than our current keyword-based methods of detecting sports and business stories in particular. There are no technological barriers preventing this, merely the issue of time and money.

3 Coding Ontologies

For several decades, two coding frameworks dominated event data research: Charles McClelland's WEIS [17, 18] and the Conflict and Peace Data Bank (COPDAB) developed by Edward Azar [1–3]. Both were created during the Cold War and

[5] Notably to traders—carbon-based and silicon-based—in the financial sector, which drives much if not most of the international reporting. The likelihood of an event being reported is very much proportional to the possibility that someone can make or lose money on it.

assumed a "Westphalian-Clausewitzian" political world in which sovereign states reacted to each other primarily through official diplomacy and military threats. While innovative when first created, these coding systems are not optimal for dealing with contemporary issues such as ethnic conflict, low-intensity violence, organized criminal activity, and multilateral intervention. McClelland [19, pg. 177] viewed WEIS as only a "first phase"; he certainly did not anticipate that it would continue to be used, with only minor modifications, for four decades.

3.1 Events

Event categories present in WEIS and COPDAB have both conceptual and practical shortcomings. For instance, WEIS has only a single subcategory for "Military engagement" that must encompass everything from a shot fired at a border patrol to the strategic bombing of cities. COPDAB contains just 16 event categories, spanning a conflict-cooperation continuum that many researchers consider inappropriate. Although there have been efforts to create alternative coding systems—most notably Leng's Behavioral Correlates of War (BCOW) [16]—WEIS and COPDAB remain the predominant frameworks in the published literature.

The lock-in of these early coding systems is readily explained by the time consuming nature of human event coding from paper and microfilm sources. Because human coders typically produce between five and ten events per hour, and a large data set contains tens of thousands of events, experimental recoding is simply not feasible. Automated coding, in contrast, allows researchers to experiment with alternative coding rules that reflect a particular theoretical perspective or interest in a specific set of issues. The effort involved in implementing a new or modified coding system, once it has been developed, is relatively small because most of the work can be done within the dictionary of verb phrases. In most cases verb phrases can be unambiguously assigned to appropriate new categories, while obscure phrases are either removed or modified. Since even a long series of texts spanning multiple decades can then be recoded in a few minutes, this allows researchers to focus on maximizing the validity of the coding scheme for their particular research program since the automated coding process itself guarantees the reliability of the system.

In the early stages of the KEDS research, we felt it was important to work with an existing framework so that we could directly compare human-coded and machine-coded data [33]. For a variety of reasons, we selected WEIS, which despite some obvious drawbacks was good enough for our initial analyses. However, we eventually decided to abandon WEIS and developed CAMEO, much as the VRA group [5, 13, 27] shifted from WEIS to the development of IDEA.

Several considerations motivated this choice. First and foremost was our long-standing concern regarding numerous ambiguities, overlaps, and gaps within the WEIS framework. In addition, the distribution of events in WEIS is quite irregular

and several of the two-digit cue categories[6] generate almost no events; we hoped we could improve on this. Third, we wanted to eliminate distinctions among actions that, while analytically discrete, could not be consistently and reliably differentiated using existing news source materials. Finally, as indicated above, the Cold War perspective that permeates WEIS makes it an inappropriate tool for studying contemporary international interactions.

Problems encountered with WEIS are exacerbated due to the lack of a fully specified standard codebook. We based our development of coding dictionaries on the version of the WEIS codebook available through the Inter-university Consortium for Political and Social Research (ICPSR) [18]. The section of the codebook dealing with event categories is quite short—about five pages—and provides only limited guidance. Since McClelland never intended that WEIS would become a de facto coding standard, the ICPSR WEIS codebook was meant to be primarily a proof-of-concept.

We initially intended CAMEO to be an extension of WEIS. Consequently, the first phase of the development of CAMEO involved adding cue and subcategories that we found theoretically necessary for the study of mediation and conflict, while keeping most of the WEIS framework intact. The next phase involved looking for examples of each category and writing definitions for the codebook. This process led to the realization that some of the distinctions we wanted to make for theoretical reasons were simply not possible given the nature of the news leads. For instance, *Promise* (WEIS 07) is almost indistinguishable from *Agree* (WEIS 08) unless the word "promise" is used in the sentence. Therefore, we eventually ended up merging the two into a single cue category—*Agree* (CAMEO 06)—that includes codes representing all forms of future positive commitment. Similarly, because verbs such as *call for, ask for, propose, appeal, petition, suggest, offer*, and *urge* are used interchangeably in news leads to refer to closely related activities, we combined *Request* and *Propose* into a single cue category—*Request/Propose* (CAMEO 05). We made similar decisions with respect to other WEIS categories such as *Grant* and *Reward*, and *Warn* and *Threaten*. We also rearranged the WEIS subcategories, both to reflect these changes and to create more coherent cue categories. As a result, *Nonmilitary demonstration* (WEIS 181) is now part of cue category *Protest* (CAMEO 14) as *Demonstrate* (CAMEO 141) while *Armed force mobilization, exercise and/or displays* (WEIS 182) is modified and falls under the new cue category *Exhibit Military Power* (CAMEO 15).

While developing CAMEO, we paid significant attention to creating a conceptually coherent and complete coding scheme. Having the cue category of *Approve* (CAMEO 03), therefore, necessitated the addition of *Disapprove* (CAMEO 11), which incorporated *Accuse* (WEIS 12) and our new addition *Protest officially* (CAMEO 113). Maintaining the cue category of *Reduce Relations* from WEIS, albeit in a modified fashion, directed us to create a parallel category that captures

[6]The phrase "cue category" refers to the broad two-digit codes, as opposed to the more specific three and four digit subcategories.

improvements in relations: *Cooperate* (CAMEO 04). In other words, we tried to insure that conceptual opposites of each cue and subcategory exist within the coding scheme, although they might not be represented by exact antonyms. We also revised or eliminated all actor-specific event codes.

In addition, we made CAMEO consistent with respect to the order of its main cue categories. Unlike WEIS and IDEA, we start with the most neutral events and move gradually from cooperation to conflict categories. While the initial coding category in WEIS and IDEA is *Yield*, CAMEO starts with *Comment* and locates *Yield* between *Provide Aid* (CAMEO 07) and *Investigate* (CAMEO 09). Technically, all three of these systems use nominal categories so that the placement of each category is irrelevant; in reality, however, the categories are often treated as ordinal or even interval variables. Therefore, CAMEO categories have an ordinal increase in cooperation as one goes from category 01 to 09, and an ordinal increase in conflict as one goes from 10 to 20.

Finally, we developed a formal codebook for CAMEO with descriptions and extensive examples for each category. We have also followed the lead of IDEA in introducing four-digit tertiary subcategories that focus on very specific types of behavior, differentiating, for instance, between agreement to, or rejection of, cease-fire, peacekeeping, and conflict settlements. These tertiary categories have been used only rarely but are available if a researcher wants to examine some very specific behaviors that might be useful in defining patterns.

Despite CAMEO originally being intended specifically to code events dealing with international mediation, it has worked well as a general coding scheme for studying political conflict. This is probably due to the fact that while CAMEO was originally going to involve a minor, 6-month revision of WEIS for a single NSF grant, we ended up spending almost 3 years on the project, with several complete reviews of the dictionaries, and hence effectively created a more comprehensive ontology.

Somewhat to our surprise, the *.verbs* dictionaries—which involved about 15,000 phrases—also needed relatively little work to produce useable data for the first phase of ICEWS. Those dictionaries had been developed for an entirely different part of the world than was coded for ICEWS, but this result was consistent with our earlier experiments in extending the data sets, which have always used a shared *.verbs* dictionary despite using specialized *.actors* dictionaries. We did one experiment where we looked at a sample of sentences where TABARI had *not* identified a verb phrase, and this produced a few new candidate phrases, but only a few. We did considerable work on cleaning up those dictionaries from the accumulated idiosyncracies of two decades of different coders, but they remained largely unchanged.

Under NSF funding, the Penn State project has made extensive efforts to re-define and generalize the entire CAMEO coding ontology using the standardized *WordNet* synsets, rather than using the current categories that were developed inductively, and these dictionaries will be available in the near future. This should help align the event coding with the larger NLP community, and probably simplify its use in languages other than English.

3.2 Actors

One of the major changes in the post-Cold War environment has been the emergence of sub-state actors as major forces in both domestic and international politics. Many commentators have argued that the proliferation of sub-state, non-state, multi-state, and trans-state actors has blurred almost completely the traditional separation of "international" and "comparative" politics. At times these groups exercise coercive force equal to or greater than that of states, whether from within, as in the case of "failed states", or across borders, as with Israel's attempts to control Hizbollah in Lebanon and Hamas in Gaza, or the near irrelevance of borders in many of the conflicts in central and western Africa. Irrespective of the effectiveness of their coercive power, these non-state actors may also be a source of identity that is more important than that of an individual's state-affiliation—the ability of al-Qaeda to attract adherents from across the Islamic world is a good example—or provide examples of strategies that are imitated across borders, as has been seen in the numerous popular revolutions in Eastern Europe or the more recent "Arab Spring."

Because they were state-centered, WEIS and COPDAB paid relatively little attention to non-state actors. A small number of long-lived opposition groups that were active in the 1960s such as the Irish Republican Army, the Palestine Liberation Organization, and the National Liberation Front of Vietnam (Viet Cong) were given state-like codes, as were major international organizations such as the United Nations and the International Committee of the Red Cross/Red Crescent. From the perspective of coding, these actors were treated as honorary states. Beyond this small number of special cases, sub- and non-state actors were ignored.

A major breakthrough in the systematic coding of sub-state actors came with the Protocol for the Analysis of Nonviolent Direct Action (PANDA) project in the early 1990s—the academic precursor to VRA and IDEA—which introduced the concept of sub-state "agents"—e.g. media, politicians, labor unions—as part of their standard actor coding. PANDA's primary focus was on contentious politics within states, and consequently needed to distinguish, for example, between police and demonstrators, or between government and opposition political parties.

Unlike PANDA, which coded the entire world, the KEDS project focused specifically on regions that have experienced protracted conflicts. As a consequence, rather than using the PANDA/IDEA approach of introducing new agent fields, we initially maintained the WEIS/COPDAB convention of using a single "source" and "target" field. However, because the areas we were coding involved quite a few sub-state actors, we eventually developed a series of standard codes that were initially a composite of the WEIS nation-state codes concatenated with PANDA agent codes. Under this system, for example, ISRMIL would be "Israel military", "LIBOPP" would be Liberian opposition parties, "SIEGOV" would be Sierra Leone government and so forth. After realizing that the simple actor-agent model did not accommodate all of the actors we wished to code, we extended this to a more general hierarchical system that was adopted, with modifications, by ICEWS.

Three principles underlie the CAMEO actor coding system. First, codes are composed of one or more three-character elements: In the present system a code can consist of one, two or three of these elements (and therefore three, six, or nine character codes), although this may be extended later. These code elements are classified into a number of broad categories, such as state actors, sub-state actor roles, regions, and ethnic groups.

Second, the codes are interpreted hierarchically: The allowable code in the second element depends on the content of the first element, and the third element depends on the second. This is in contrast to a rectangular coding system, where the second and third elements would always have the same content. The most familiar analogy to a hierarchical coding system is the Library of Congress cataloguing system, where the elements of the catalog number vary—systematically—depending on the nature of the item being catalogued, and consequently may contain very different information despite being part of a single system. The event coding system used in BCOW [16] is another example of a hierarchical scheme in the event data literature.

Third, we are basing our work on standardized codes whenever these are available. This is most obvious in our use of the United Nations nation-state codes (ISO-3166-1 ALPHA 3) (http://unstats.un.org/unsd/methods/m49/m49alpha.htm). This contrasts to the Russett-Singer-Small codes [29] used in WEIS, which are specific to the North American quantitative international relations community. We have generally adopted the IDEA agent codes for sub-state actors. We originally used the HURIDOCS (http://www.huridocs.org/) classifications for world religions, but subsequently expanded this to the much more comprehensive and systematic list found in the CAMEO "Religious Classification System." (http://eventdata.psu.edu/cameo.dir/CAMEO.0.10b2.pdf; this same source also provides a standard set of ethnic codes which we developed by comparing a number of existing sets of ethnicity and languages codes, though we primarily based this on the Joshua Project (http://www.joshuaproject.net/) and Ethnic Power Relations (http://www.epr.ucla.edu/) typologies.

In the later phases of the ICEWS project, Lockheed also developed substate agent typologies which provided considerably more detail than that provided in the classical coding schemes; details on this system and the various proprietary software developed to support it can be found in [36]. Lockheed's system integrates the coding scheme with a large database of group characteristics and allows for the rapid customization of coding schemes.

Unfortunately, standard codes are generally not available. For example, most IGOs are known by acronyms of varying lengths, so we need to decide how to truncate these to three characters. We spent considerable time trying to determine whether the U.S. government had a standard list of militarized non-state actors; as best we can tell, this does not exist (or at least not in a form we can access), and the situation for ethnic groups is similar.

4 Actor Dictionaries and Named Entity Recognition

By far the greatest challenge of scaling-up the KEDS/TABARI system has been in the area of actor dictionary development. The KEDS project had focused on a small number of geographical areas, primarily the Levant, with 10-year data sets on the Balkans and West Africa. We had done some experimental work under small government contracts to code individual countries in other areas of interest, in all parts of the world, for short—typically 2-year—time periods, and graduate student research by Ömür Yilmaz and Baris Kesgin had produced very detailed dictionaries for Turkey, but that was it. ICEWS, in contrast, initially involved coding 29 states that encompass more than half the world's population, and in the final stages was expanded to coding the entire world.

The earlier KEDS data sets were initially developed by individuals—largely undergraduate honors students—who went through sentences item by item and added new patterns to the actor and verb dictionaries as they encountered incorrectly coded sentences.[7] This was later supplemented by a relatively simple named-entity-recognition (NER) program called *ActorFilter* that would locate potential new names based on capitalization patterns, compare these to entries in the existing dictionaries, and then produce a keyword-in-context (KWIC) listing of entities which appeared to be new, listed in reverse order of frequency. This was particularly useful in making sure that any major new actors were not missed, and was our first step in developing dictionaries for new countries.

Neither of these techniques scaled, particularly in the relatively short time frame of the first phase of the ICEWS work. While we did some spot-checking of individual stories, our ability to do this with any meaningful proportion of the 26-million sentences in the ICEWS corpus was limited. *ActorFilter*, unfortunately, had not been designed for a project of this magnitude and while it could be used on a sample, it slowed to an unusable crawl on very large files.

Consequently, three approaches were used.

First, rather than deriving the actors from the texts, we tried to locate lists of actors and incorporate these into both international and nation-specific dictionaries. Various national sources provided lists of parliamentarians and other local leaders, and we've also been expanding the list of NGOs and IGOs. As a consequence, the Asian actors dictionaries now have around 20,000 entries, compared to the 1,000 or so entries typical in earlier KEDS work.

We also augmented a reference file used in earlier NSF-funded work on the Militarized Interstate Disputes dataset [35] with information in the *CIA World Factbook* and rulers.org to a comprehensive list of state names, major cities, regions

[7]To date, all of the successful automated event data coding systems are dictionary and rule based, rather than using statistical-methods: see [36]. While statistical methods would certainly be attractive, and seem to work on highly simplified "toy problems" such as those in [6], all of the successfully-deployed systems to date are dictionary-based, and numerous efforts to scale initially-promising statistical methods have failed.

and geographical features, adjectival forms, and date-delimited lists of heads of state and other members of government. This has developed into the roughly 32,000-entry *CountryInfo* (http://eventdata.psu.edu/software.dir/dictionaries.html) which has a systematic format fairly close to that of XML, and can easily be converted into TABARI dictionary format with a utility program.

Second, we improved the ability of TABARI to automatically assemble codes from combinations of a named actor and an generic agent; this facility is also part of JABARI-NLP. For example "Philippine soldiers" will automatically generate the code `PHLMIL`, whereas "The Philippine Secretary of Agriculture" will automatically generate the code `PHLGOV`. Earlier dictionaries had done this directly, with separate dictionary entries for, say, "Australian police," "Cambodian police," "Chinese police" and so forth. The new system is both faster in terms of the dictionary size and much more efficient. This allows the coding of both generic agents such as "police", "soldiers", "demonstrators" and the like, as well as named individuals where we have the title in the dictionary but not the individual person. For most of our coding, at least for the forecasting efforts in ICEWS, individual identities are not used, so this gets quite a bit of information we were previously missing. In support of this new facility, we also increased the size of the *.agents* dictionary considerably, based on *WordNet* and sampling from the source texts.

Finally, *ActorFilter* was replaced with a new open-source `Python` program, `PoliNER`, which had a similar function but was adapted to the much larger dictionaries and source text files. The sorted output of this program can be combined with a program named `CodeCatcher` for machine-assisted development of dictionaries: `CodeCatcher` guesses the likely code based on known entities in a sentence, and allows rapid combination of codes based on that other information.

These efforts were a major step forward, but dictionary development—and maintenance, as dictionaries need to be updated as political figures change—remains a considerable challenge. Fortunately there is a considerable literature—much of it DARPA-funded—on NER, and some of these methods are very sophisticated—for example using conditional random fields and hidden Markov models—and are certainly far more sophisticated than what we are currently using, and these methods might provide significant additional advances in efficiency.

5 Pre-processing Using NLP Tools

A major shift in automated coding that has been shown to dramatically increase accuracy has been the incorporation of open-source natural language processing (NLP) tools to correctly identify the elements of the sentence required for coding. When KEDS was being developed in the early 1990s, or even in the early 2000s, the development period of TABARI, open-source code was still a relative novelty. As a consequence, these programs handled all of their own linguistic processing

with an internal shallow parser written into the code. Parser code written by a political scientist. This obviously *worked*, in the sense of producing useable data, but the internal structure of the program is quite complex and difficult to modify. In the environment of the 2010s, it makes far more sense to leave NLP software development to the computational linguists, and focus only on those remaining tasks that are needed to get convert these structures to events.

This is the approach that was taken with JABARI-NLP. The original JABARI simply duplicated TABARI in a Java environment [38].[8] However, after several key weaknesses were identified in the shallow-parsing approach—most importantly, a tendency to match words in verb phrases that were not actually part of the phrase—the JABARI effort, rather than attempting to deal with these in the program itself, explored a number of open-source options that could provide the NLP processing, then was modified to handle that information. TABARI is gradually being modified in a similar fashion.

For purposes of illustration, consider the following initial sentences for a news story:

> US Supreme Court Justice Stephen Breyer was robbed by a machete-wielding man at his Caribbean vacation home, a Supreme Court spokeswoman said.
> The robber broke into Judge Breyer's home on the island of Nevis around 21:00 EST (02:00 GMT) on Thursday.
> The Supreme Court justice was at home with his wife and guests, but no one was hurt, the spokeswoman said.

Software for the following tasks can be found at open-source NLP software site such as Open-NLP and various other academic sites; we are going to discuss these generally by function rather than making specific recommendations, since this is still very much an evolving field.

- Sentence delineation. As noted in Sect. 2, this is a surprisingly difficult task given the presence of abbreviations, punctuation occurring inside sentences, and the occurrence of character strings that are not actually part of the sentence, particularly across multiple story formats. Linguists have systems that are more robust than our `perl` filters.
- Disambiguation by parts-of-speech markup. One of the major tasks of the TABARI dictionaries is noun-verb disambiguation: this issue accounts for much of their size. Parts-of-speech (POS) marking—or in the example below, a system that makes noun-verb distinctions and also classifies these into general categories—would eliminate this problem.

```
US/noun.group Supreme/noun.group Court/noun.group
    Justice/noun.group
Stephen/noun.person Breyer/noun.person was
    robbed/verb.possession by
```

[8] Including, at the request of the sponsor, some bugs in TABARI, though after the equivalence of the two systems was demonstrated, these were corrected in both systems.

```
a machete-wielding man/noun.person at his/pronoun
    Caribbean
vacation/noun.artifact home/noun.artifact,
    a Supreme/noun.group
Court/noun.group spokeswoman/noun.person
    said/verb.communication.

The robber/noun.person broke/verb.communication
    into/verb.communication
Judge/noun.person Breyer/noun.person's
    home/noun.location on the
island/noun.object of Nevis/noun.location around
    21:00 EST/noun.time on
Thursday/noun.time.
```

- Stemming. TABARI has only recently added capabilities of automatically recognizing the regular forms of nouns and verbs. Many NLP systems use stemming—most frequently the Porter stemming algorithm for English (http://tartarus.org/martin/PorterStemmer/). This should both simplify and generalize the dictionaries.

- Full parsing. An assortment of full-parsers—as distinct from the shallow parsers used in KEDS/TABARI—are available, and the *TreeBank* parse format appears to be a fairly stable and standard output format. This allows a researcher to use the parser of his or her choice (notably some parser developed in the future) so long as these could produce *TreeBank*-formatted output. The most important contribution of the full parsing is insuring that the words associated identified as belonging to a verb phrase are in fact associated with that verb, and not with a subordinate clause or some other part of the sentence.

```
(ROOT (S (S (NP (NNP US) (NNP Supreme) (NNP Court)
    (NNP Justice)
(NNP Stephen) (NNP Breyer)) (VP (VBD was)
    (VP (VBN robbed) (PP (IN by)
(NP (NP (DT a) (JJ machete-wielding) (NN man))
    (PP (IN at) (NP (PRP$ his)
(JJ Caribbean) (NN vacation) (NN home)))))))) (, ,)
    (NP (DT a)
(NNP Supreme) (NNP Court) (NN spokeswoman))
    (VP (VBD said)) (. .)))
```

- Pronoun and entity coreferencing. Some of the full-parsing systems provide pronoun and entity coreferencing, another feature coded into TABARI. Alternatively, this can be provided in stand-around coreferencing systems such as the ARK noun phrase coreferencer. (http://www.ark.cs.cmu.edu/ARKref/)

```
<ref id="1" ent="1_4_8">US Supreme Court Justice
    Stephen Breyer</ref> was
robbed by <ref id="2" ent="2">a machete-wielding
    man at
<ref id="3" ent="1_4_8">his</ref> <ref id="4"
    ent="3_7_46">Caribbean vacation
home</ref>, <ref id="5" ent="5_21">a Supreme Court
    spokeswoman</ref> said.

<ref id="6" ent="6_19">The robber</ref> broke into
    <ref id="8" ent="1_4_8">
Judge Breyer's</ref> <ref id="7" ent="3_7_46">
    home</ref> on
<ref id="9" ent="9">the island of Nevis</ref>
    around 21:00 EST on
<ref id="13" ent="13">Thursday</ref>.

<ref id="17" ent="1_4_8">The Supreme Court
    justice</ref> was at home with
<ref id="19" ent="1_4_8">his</ref> wife and guests,
    but <ref id="20" ent="20">
no one</ref> was hurt, <ref id="21" ent="5_21">the
    spokeswoman</ref> said.
```

The use of these tools accomplishes at least the following improvements:

- It aligns automated event coding—which is fundamentally an NLP problem—with the larger NLP community. As their tools improve, we can incorporate those improvements into event data work immediately.
- It considerably simplifies—though not entirely eliminating the need for—the construction and maintenance of coding programs, and in particular the tasks that can now be done with open-source ancillary programs would eliminate many of the most brittle parts of the original TABARI code.
- It introduces a deep—as distinct from a shallow—parser into the system, and the shallow parsing approach has probably reached its limits.
- The use of standardized NLP tools and dictionaries would probably simplify the development of a system for languages other than English, particularly languages such as Chinese and Arabic where considerable NLP work has been invested;
- Many of these features should simplify the *.verbs* dictionaries, or at the very least gain more robust performance from dictionaries of the same length;

Parsing and other pre-processing—in all likelihood a fairly slow process—needs to be done only once for a given sentence, and the marked-up version can be stored, so unlike systems with in-line deep parsers, the resulting coding (which is likely

to be re-done many times) should be as fast or faster than the current system. The pre-processing is also trivially divided across multiple processors in a cluster system, so with suitable hardware or using virtual clusters in a cloud computing environment, the processing requirements can be easily adjusted to near-real-time coding environments.

6 Coding and Post-processing

6.1 Cluster Processing

TABARI is an open-source C++ program—compiled under `gcc`—that runs on a common code base in both the Macintosh OS-X and various Linux/Unix environments. This has proved useful in deploying it across a combination of desktop, server and cluster environments.[9]

The major innovation in conjunction with the 2009 coding for the second phase of ICEWS was the use of a computer cluster to dramatically increase the coding speed. In the 2008 data development for ICEWS Phase I, coding the 1997–2004 data on personal computers required almost a week. This was also slowed by the existence of some bugs in TABARI that occurred only with extremely rare sentence structures and thus had gone undetected in earlier work with the program: there were initially eight of those out of the 26-million sentences.

In 2009, we gained access to a small, 14-processor cluster computer that was sitting unused (and undocumented) at the University of Kansas. Rather than trying to get TABARI to run in parallel at the micro level, we did "parallelism on the cheap" and simply split the text files to be coded across the processors, which shared a common file space, coded these simultaneously, then re-combined the output files at the end of the run. TABARI ran on the individual nodes at around 5,000 sentences per second; the throughput for the cluster as a whole ended up around 70,000 stories per second, allowing the entire 26-million story corpus to be coded in about 6 min. The initial set-up, of course, took quite a bit longer, but this was particularly useful for weeding out the aforementioned problematic records that would cause the program to crash.

A 14-processor cluster is, of course, tiny—Penn State has multiple clusters available to social scientists that are in the 256-processor range—so effectively the coding speed is unlimited, even for a very large corpus. Furthermore, this can be done by simple file spliting, so the gain is almost linear.

[9] In principle these enhancements could also be applied to JABARI-NLP, though it is running in secure military systems rather than open environments and to date has made less use of cluster processing.

6.2 One-A-Day Filtering

Following the protocols used in most of the research in the KEDS project, the major post-processing step is the application of a "one-a-day" filter, which eliminates any records that have exactly the same combination of date, source, target and event codes. This is designed to eliminate duplicate reports of events that were not caught by earlier duplicate news report filters. In our work on the Levant data set, this fairly consistently removes about 20 % of the events; the effect on the ICEWS data may be somewhat higher due to the use of a greater number of sources.

In areas of intense conflict—where multiple attacks could occur within a single dyad in a single day—this could eliminate some actual events. However, these instances are rare, and periods of intense conflict are usually obvious from the occurrence of frequent attacks across a month (our typical period of aggregation), and do not require precise measures within a single day. Periods of intense conflict are also likely to be apparent through a variety of measures—for example comments, meetings with allies, offers of aid or mediation—and not exclusively through the attacks themselves.

6.3 Sophisticated Error Detection/Correction

Thus far, we have been using only limited error detection and correction. Some LM-ATL experiments have shown that even very simple filters focusing on anomalous high-intensity events can eliminate egregious errors such coding USA/Japanese conflict events based on Pearl Harbor travel and movie reviews or anniversaries of the bombings of Hiroshima and Nagasaki. Eliminating these is particularly important when the output is used for the monitoring of unlikely events—for example pattern recognition of potential conflict "triggers" either by humans or machine-learning algorithms—as distinct from conventional statistical approaches which can readily ignore these as noise. In addition, far more sophisticated filtering methods are available, and many of these are of relatively recent vintage due to the computing power required. A multi-category support vector machine (SVM), for example, could be applied to the full text of a story—or possibly a single sentence, but SVMs tend to work better at the document level than the sentence level—to determine whether the story is likely to have produced events of the type coded, based on previously verified correct codings.

From this point, a variety of different things are done with the data, but these fall into the category of data management and model construction, rather than data generation per se. LM-ATL [36] is developing an increasingly elaborate system for the management of the data that includes a wide variety of visualization tools, as well as interactive "drill-down" capability that allow a user to go from the coded events back to the original text, as well as management and display of the coding

dictionaries. On the modeling side, the data can be aggregated in a variety of ways, including event counts for various types of dyads as well interval-level scaled data using a modification of the Goldstein scale [11] for the CAMEO ontology.

7 Open Issues

7.1 Geolocation

A still missing component of the system is the ability to tag the entire story with the location, which will allow the agents to be coded even if they are not preceded by a national identifer. This is particularly important in local sources: unlike an international news report, a Philippine news report on Mindanao, for example, will almost never mention that Mindanao is part of the Philippines. There are several software systems for doing this type of tagging and LM-ATL is experimenting with them [36] with some success, though this is still an open issue. As with NLP processing more generally, this is an open research area with a variety of active open-source and proprietary systems available, and is likely to improve substantially in the near future.

7.2 Machine Translation

With the increasing availability of news items in multiple languages on the web—for example European Media Monitor looks at sources in 43 languages—the possibility of coding in languages other than English is very attractive. There are at least three different approaches that could be used here.

The most basic, but by far the most labor intensive, would be to simply write an equivalent automated coding system for other languages, and come up with equivalent *.verbs* dictionaries. The *.actors* dictionaries would probably require little modification for languages using the Latin alphabet; though they would require extended work for systems such as Arabic, Chinese and Hindi. We did this for German in an early phase of the KEDS project [9], albeit with very simple dictionaries. While some modification of the parser is required in this approach, shallow parsing looks at only the major syntactic elements of a sentence and this would be relatively easy, and the *linguistic* work of Noam Chomsky strongly suggests that this modifications will fall into a relatively small number of categories.

The second possibility would be to use NLP tools to handle the parsing—which we are likely to be doing in the next phase of the development of the English-language coders as well—but still use language-specific *.verbs* dictionaries. The modification of the *.verbs* dictionaries would also allow language-specific idiomatic

phrases—which are likely to be quite important and quite unsystematic—but would also involve considerable work. This might, however, be justified in the cases of languages where there is a large set of news sources, particularly on local events, which is not covered well in English: Spanish and Arabic come to mind, as would Chinese if an independent press develops in that country.

The final possibility, which was pursued at an experimental level by Lockheed [36], is to use machine translations into English of the source texts, and then continue to use the English-language coders. The extent to which this works depends both on the quality of the automated translators, and the extent to which the existing dictionaries—generally developed on texts at least edited if not written by fluent writers of English—correspond to the phrases encountered in the automated translations, which are often based on statistical methods intended simply to provide a recognizable sense of the text, not an eloquent rendition of it.

Lockheed's initial experiments with several translation systems working on roughly two-million sentences in Spanish and Portuguese achieved accuracy around 67 %, which is probably comparable to human coding accuracy and would provide useful data for statistical modeling but this is not sufficiently high to satisfy human users working with the data at a highly detailed level [25]. There has been extensive work in machine translation into English from Spanish, Arabic, and Chinese, and as with the other NLP tools, these systems are likely to improve over time given the economic motivations for developing good software.

7.3 Real-Time Coding

At Kansas during the 2009–2010 period we undertook an experiment in true real-time coding using RSS feeds. RSS feeds present a potentially very rich source of real-time data because they are available in actual real time using standard software, and, of course, are free. The downside of RSS feeds is the absence—at least at the present time—of any archival capacity, so they can be used for current monitoring but not for generating a long time series.

A variety of RSS feeds are available. The richest would be two major RSS aggregators, GoogleNews and European Media Monitor, which track several thousand sources each. In some experimental downloads in 2008, we found that these generated about 10 Gb of text per month, and that volume has probably only increased. The two downsides with the aggregators are massive levels of duplication, and the fact that they are not produced in a standard format: instead, each source must be reformatted separately. This is not particularly difficult in terms of simply detecting the natural language text of the news report itself—and in fact all of these feeds consist largely of HTML code, which typically takes up more than 90 % of the characters in a downloaded file—but can be difficult in terms of detecting dates and sources.

Instead of looking at the aggregators, we focused on two high-density individual sources: Reuters and UPI. In addition to providing RSS feeds, these also have archives, back to 2007 for Reuters and back to 2001 for UPI; these could be downloaded from the Web. The focus on individual sources meant that only a small number of formats had to be accommodated—even formats within a single source exhibit some minor changes over time—but these two sources, as international news wires, still provide relatively complete coverage of major events. They do not, however, provide the same level of detail as the commercial sources, Factiva for Reuters and LN for UPI. After some experimentation, it turned out to be easier to access the updates to this information from their web sites rather than through RSS feeds per se, but this still allows fairly rapid updating.

Implementation of a real-time coder was a relatively straightforward task of linking together, on a server, the appropriate reformatting and duplicate detection programs, running TABARI at regular intervals on the output of those programs, and then storing the resulting event data in a form that could be used by other programs: mySQL was used for this purpose. While the basic implementation of this system has been relatively straightforward, our 18-month experiment found at least three characteristics of the data that should be taken into account in the design of any future systems.

First, while *in principle* one could get real-time coding—automated news monitoring services used in support of automated financial trading systems routinely do this—there is little reason to do so for existing event data applications, which generally do not work on data that is less finely grained than a day. Furthermore, the news feeds received during the course of a day are considerably messier—for example with minor corrections and duplications—than those available at the end of a day. Consequently, after initial experiments we updated the data only once a day rather than as soon as the data became available.

Second, these are definitely not "build and forget" systems due to the changing organization of the source web sites. Reuters in particular has gone through three or four major reorganizations of their web site during the period we have been coding data from it, and in one instance was off-line for close to a week. Thus far, the changes in code resulting from these reorganizations have been relatively minor, primarily dealing with the locations of files rather than the file formats, but it has necessitated periodic—and unexpected—maintenance. The RSS feeds may have been more reliable—these presumably did not go off-line for a week—but still probably undergo some changes. It is also possible that as the sites mature, they will be more stable, but this has not occurred yet.

Finally, we have not dealt with the issue of automatically updating actor dictionaries, depending instead on general international dictionaries that contain country-level information but relatively little information on individual leaders. International news feeds generally include national identification—"United States President Obama," not just "Obama"—so the country-level coding should generally be accurate, but the data probably is less detailed at the sub-state level.

8 Conclusion

In a history of the first 15 years of the KEDS/TABARI project [31], the final section—titled "Mama don't let your babies grow up to be event data analysts" lamented the low visibility of event data analysis in the political science literature despite major advances in automated coding and the acceptance of analyses resulting from that data in all of the major refereed political science journals.

The situation at the present is very different, largely due to ICEWS, which emerged about 6 months after that history was written. All three of the teams involved in the first phase of ICEWS used some form of event data in their models. Lockheed, the prime contractor for the only team whose models cleared the out-of-sample benchmarks set by ICEWS, has continued to invest in additional developments, both for ICEWS and potentially for other projects, and as noted in the previous section, there are now a number of proprietary systems in active development, in contrast to the previous 15 years which saw only KEDS/TABARI and VRA-Reader. At the same time, there has been substantial NSF funding of further development of the open-source TABARI and various ancillary utilities, so while the open-source work lags somewhat behind the proprietary—though in other aspects, such as the incorporation of *WordNet* into the dictionaries, it is ahead—reasonably up-to-date software is available as open source, and it is still being actively developed.

In 1962, Deng Xiaoping famously quoted the Sichuan proverb, "No matter if it is a white cat or a black cat; as long as it can catch mice, it is a good cat." Statistical models utilizing event data coded with automated techniques are good cats. Some are white, some are black, but they catch mice. Furthermore, the fact that such models exist is now known [25, 26] and from a policy perspective it is likely that they will be continued to be developed for policy applications seems rather high: the open-access textbook on the results of the KEDS project circa 2000, *Analyzing International Event Data,* reportedly has been translated into Chinese.[10] The cat, so to speak, is out of the bag.

Acknowledgements This research was supported in part by contracts from the Defense Advanced Research Projects Agency under the Integrated Crisis Early Warning System (ICEWS) program (Prime Contract #FA8650-07-C-7749: Lockheed-Martin Advance Technology Laboratories) as well as grants from the National Science Foundation (SES-0096086, SES-0455158, SES-0527564, SES-1004414) and by a Fulbright-Hays Research Fellowship for work by Schrodt at the Peace Research Institute, Oslo (http://www.prio.no). The results and findings in no way represent the views of Lockheed-Martin, the Department of Defense, DARPA, or NSF. It has benefitted from extended discussions and experimentation within the ICEWS team and the KEDS research group at the University of Kansas; we would note in particular contributions from Steve Shellman, Hans Leonard, Brandon Stewart, Jennifer Lautenschlager, Andrew Shilliday, Will Lowe, Steve Purpura, Vladimir Petroff, Baris Kesgin and Matthias Heilke.

[10]Though we've not been able to locate this on the web. Itself interesting.

References

1. Azar EE (1980) The conflict and peace data bank (COPDAB) project. J Confl Resolut 24: 143–152
2. Azar EE (1982) The codebook of the conflict and peace data bank (COPDAB). Center for International Development, University of Maryland, College Park
3. Azar EE, Sloan T (1975) Dimensions of interaction. University Center for International Studies, University of Pittsburgh, Pittsburgh
4. Bond D, Bond J, Oh C, Jenkins JC, Taylor CL (2003) Integrated data for events analysis (IDEA): An event typology for automated events data development. J Peace Res 40(6): 733–745
5. Bond D, Jenkins JC, Taylor CLT, Schock K (1997) Mapping mass political conflict and civil society: Issues and prospects for the automated development of event data. J Confl Resolut 41(4):553–579
6. Boschee E, Natarajan P, Weischedel R (2012) Automatic extraction of events from open source text for predictive forecasting. In: Subrahmanian V (ed) Handbook on computational approaches to counterterrorism. Springer, New York
7. Chenoweth E, Dugan L (2012) Rethinking counterterrorism: evidence from israe. Working Paper, Wesleyan University, Middletown, CT
8. Dugan L, Chenoweth E (2012) Moving beyond deterrence: the effectiveness of raising the expected utility of abstaining from terrorism in israel. Working Paper, University of Maryland, College Park, MD
9. Gerner DJ, Schrodt PA, Francisco RA, Weddle JL (1994) The machine coding of events from regional and international sources. Int Stud Q 38:91–119
10. Gleditsch NP (2012) Special issue: event data in the study of conflict. Int Interact 38(4): 375–569
11. Goldstein JS (1992) A conflict-cooperation scale for WEIS events data. J Confl Resolut 36:369–385
12. Howell LD (1983) A comparative study of the WEIS and COPDAB data sets. Int Stud Q 27:149–159
13. Jenkins CJ, Bond D (2001) Conflict carrying capacity, political crisis, and reconstruction. J Confl Resolut 45(1):3–31
14. Kahneman D (2011) Thinking fast and slow. Farrar, Straus and Giroux, New York
15. King G, Lowe W (2004) An automated information extraction tool for international conflict data with performance as good as human coders: A rare events evaluation design. Int Organ 57(3):617–642
16. Leng RJ (1987) Behavioral correlates of war, 1816–1975. (ICPSR 8606). Inter-University Consortium for Political and Social Research, Ann Arbor
17. McClelland CA (1967) World-event-interaction-survey: a research project on the theory and measurement of international interaction and transaction. University of Southern California, Los Angeles, CA
18. McClelland CA (1976) World event/interaction survey codebook (ICPSR 5211). Inter-University Consortium for Political and Social Research, Ann Arbor
19. McClelland CA (1983) Let the user beware. Int Stud Q 27(2):169–177
20. Merritt RL, Muncaster RG, Zinnes DA (eds) (1993) International event data developments: DDIR phase II. University of Michigan Press, Ann Arbor
21. Mikhaylov S, Laver M, Benoit K Coder reliability and misclassification in the human coding of party manifestos. Political Anal 20(1):78–91 (2012)
22. Mooney B, Simpson B (2003) Breaking News: How the Wheels Came off at Reuters. Capstone, Mankato
23. Nardulli P (2011) The social, political and economic event database project (SPEED). http://www.clinecenter.illinois.edu/research/speed.html

24. Nardulli PF, Leetaru KH, Hayes M Event data, civil unrest and the SPEED project (2011). Presented at the International Studies Association Meetings, Montréal
25. O'Brien S (2012) A multi-method approach for near real time conflict and crisis early warning. In: Subrahmanian V (ed) Handbook on computational approaches to counterterrorism. Springer, New York
26. O'Brien SP (2010) Crisis early warning and decision support: contemporary approaches and thoughts on future research. Int Stud Rev 12(1):87–104
27. Petroff V, Bond J, Bond D (2012) Using hidden Markov models to predict terror before it hits (again). In: Subrahmanian V (ed) Handbook on computational approaches to counterterrorism. Springer, New York
28. Ruggeri A, Gizelis TI, Dorussen H (2011) Events data as bismarck's sausages? intercoder reliability, coders' selection, and data quality. Int Interact 37(1):340–361
29. Russett BM, Singer JD, Small M (1968) National political units in the twentieth century: a standardized list. Am Political Sci Rev 62(3):932–951
30. Schrodt PA (1994) Statistical characteristics of events data. Int Interact 20(1–2):35–53
31. Schrodt PA (2006) Twenty years of the Kansas event data system project. Political Methodol 14(1):2–8
32. Schrodt PA (2012) Precedents, progress and prospects in political event data. Int Interact 38(5):546–569
33. Schrodt PA, Gerner DJ (1994) Validity assessment of a machine-coded event data set for the Middle East, 1982–1992. Am J Political Sci 38:825–854
34. Schrodt PA, Gerner DJ, Yilmaz Ö (2009) Conflict and mediation event observations (CAMEO): an event data framework for a post Cold War world. In: Bercovitch J, Gartner S (eds) International conflict mediation: new approaches and findings. Routledge, New York
35. Schrodt PA, Palmer G, Hatipoglu ME (2008) Automated detection of reports of militarized interstate disputes using the SVM document classification algorithm. Paper presented at American Political Science Association, Chicago, IL
36. Shilliday A, Lautenschlager J (2012) Data for a global icews and ongoing research. In: 2nd international conference on cross-cultural decision making: focus 2012, San Francisco, CA
37. Tetlock PE (2005) Expert political judgment: how good is it? how can we know? Princeton University Press, Princeton
38. Van Brackle D, Wedgwood J (2011) Event coding for hscb modeling: challenges and approaches. In: Human social culture behavior modeling focus 2011, Chantilly, VA

http://www.springer.com/978-1-4614-5310-9

Handbook of Computational Approaches to Counterterrorism

Subrahmanian, V.S. (Ed.)
2013, XVIII, 578 p., Hardcover
ISBN: 978-1-4614-5310-9