Automated Coding of Very Large Scale Political Event Data

Philip A. Schrodt and Jay Yonamine

Abstract This paper discusses the current state-of-the-art for generating high-volume, near-real-time event data using automated coding methods, based on recent efforts by the Penn State Event Data Project and its precursors. Political event data—the extraction of who did what to whom based from natural language news reports—is a core element of many models that forecast political stability, and is easily adapted to new problems, either in a fully-automated mode or as part of a machine-assisted coding system. We discuss five elements of a contemporary system: acquisition and filtering of source texts, the actor and event coding ontology, the development of actor dictionaries using named-entity recognition, the development of a coding engine, and integration with open-source natural language processing software. While we will be focusing on the characteristics of the open-source TABARI, most of these issues will be relevant to any system, and thus is intended as a "how-to" guide to the pragmatic challenges and solutions to various elements of the process of generating event data using automated techniques.
Version 0.8: September 27, 2012

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 [16] as well as a variety of more specialized efforts such as Leng's BCOW [14]. By the late 1980s, the NSF-funded *Data Development in Inter-*

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

Jay Yonamine
Political Science, Pennsylvania State University, University Park, PA 16801, USA. e-mail: jxy190@psu.edu

national Relations project [18] 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, 27, 25]) and the proprietary VRA-Reader (`http://vranet.com`; [13]) and in the 2000s, the development of two new political event coding ontologies—CAMEO [28] and IDEA [5]—designed for implementation in automated coding systems. The U.S. Department of Defense Intergrated Conflict Early Warning System (ICEWS) project made extensive use of event data in the development of forecasting models for Asia; details can be found in [22]. A summary of the current status of political event projects, as well as detailed discussions of some of these, can be found in [26, 10].

The purpose of this paper is to describe a number of incremental improvements and lessons-learned in the recent experience of both our open-source work at Kansas and Penn State.This paper 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 paper essentially goes through the various phases of a machine-coding project, outlined schematically in Figure 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.

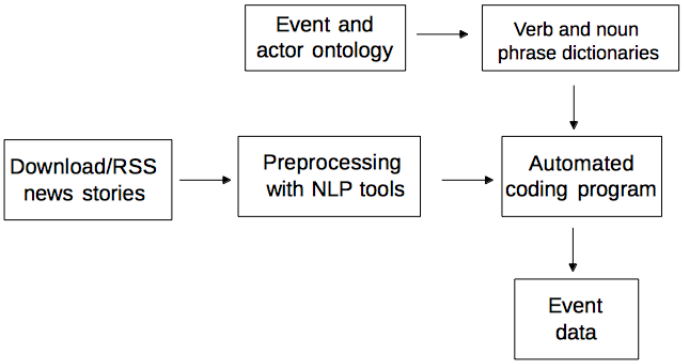

Fig. 1 Process of Generating Event Data by Automated Methods

From the outset, we would emphasize that automated coding is still a work in progress. It has clearly crossed the threshold into the realm of practical utility 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 (NLP), and advances in that area will undoubtedly substantially

increase the accuracy of our methods, and have begun to 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 and TABARI—and some of these issues will differ for coding systems using alternative approaches.

1.1 Human versus machine coding

In some circles, automated coding and statistical forecasting can be a tough 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 [12, 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. Any event data system, be it human- or machine-coded, will have plenty of those. However, multiple published tests [13, 27] have shown that machine coding is comparable in accuracy to human coding, even though the human coding accuracy in some of those tests is quite low: King and Lowe [13] 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, a recent analysis [19] has shown 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%. Human coding is anything but flawless.

Unfortunately, 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,[1] but based on 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,

[1] The *major* problem with such an exercise is reaching convergence among the human coders: about ten 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.

Russ Leng's BCOW) over the years, sustained accuracy tends to be in the range of 70% at best. The human-coded COPDAB data set somehow manages to miss the Korean War [11], the human-coded GEDS project, which consumed to 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.

Arguments for the superiority of human coding also ignore the fact that the *total* amount of information available to a system using fully automated methods is vastly greater than that which can be processed by an individual. While the intuitive analysis of a skilled human analyst or coder may be better in an individual case (and certainly in the assessment of an individual news report), the *composite* of the automated system almost certainly has better performance. A subject-matter-experts (SMEs) may perform better on their area of expertise in a particular time frame (though Tetlock's research [30] would suggest not even this is true), but there is little evidence that they can perform broadly. In contrast, using event data, recent event data efforts are being scaled to provide daily level event for the entire world over a substantial period of time.

As noted in [24], if one is using event data in forecasting models—the objective of many 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

1. 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;
2. Event ontologies such as WEIS, CAMEO and IDEA (discussed thoroughly in Section 3) are very generic and bin together events that may not always belong together in all contexts;
3. Forecasting models always contain specification error and cannot consider everything; for example few if any political forecasting models contain a full economic forecasting component;
4. 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.

An additional problem exist when assessing the alternative of human coding for generating event data: simple impossibility. Current automated event data projects have intakes of 20,000 to 100,000 stories per day. Even assuming that 80% of these are duplicates or irrelevant and could be removed completely efficiently, the remaining coding at the upper level would require a permanent team of around 100 to 400

coders—depending on the effectiveness of the machine-assisted coding software—a labor requirement that would probably need to be multiplied by at least 1.5 to account for management, training, quality control and turnover. This is simply beyond the capacity of existing academic political science enterprises. Alternatively, TABARI codes about 5,000 sentences per second, and any automated coder can be trivially scaled to any speed needed by dividing the texts across multiple processors in now widely-available cluster computers and thus presents no such problems. For this reason, human-machine comparisons are of little practical consequence, since human coding is generally not an option.

Table 1 Tradeoffs between human and automated coding

Attributes of human coding	**Attributes of machine coding**
•Best for small data sets	•Best for large data sets
•Slow	•Fast
•Data coded one time at a single site	• Data may be coded over a period of time or across institutions
•Non-replicable	•Perfect replicability
•No relevant dictionaries	•Existing dictionaries can be modified
•Native speakers can code complex sentence structures	•Best coding simple sentence structures
•Native speakers can interpret metaphorical, idiomatic, or time-dependent text	•Best at coding literal, present-tense text
•Expensive to pay coders, trainers and supervisors	•Near zero costs after initial development
•Errors often subjectively correct	•Errors often objectively incorrect

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*. For example, the Chenoweth and Dugan project [6, 8] has used TABARI as a sophisticated prefilter for coding incidents of terrorism, with a substantial reduction in the required labor costs.

2 Text acquisition and formatting

Returning to the event data generation process, the first step is the acquisition of news reports to code. Following the lead of most event data projects, we have relied primarily on the Lexis-Nexis (LN) and Factiva data services. Both, unfortunately, require a tedious process of manual downloading.[2] In the future, however, we expect that projects will use a combination of bulk downloads—*surely* at some point LN and Factiva are going to catch up with the fact that there is a legitimate role for very large scale text mining and provide reasonably priced alternatives to manual downloads[3]—and in any case, real-time data will be provided by some combination of RSS feeds, automated monitoring of web sites for new content, and frequent downloads from dedicated servers using APIs designed for this task.

The use of multiple sources provides a challenge in extracting the required information—the date, source and individual sentences—from the original download. To date, we have largely been using source-specific filters, generally in `perl`. While LN and Factiva tend to be 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. In contrast, Reuters—possibly because it is widely used in fully automated financial applications, which are likely to drive considerable standardization in the future—has a very clean format that is very simple to parse.

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 cannot depend on this, particularly for the regional sources that may or may not use a consistent reporting format. These exceptions are sufficiently 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 to 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 expression that will identify non-material: For example something of the form `\d+\-\d+` is almost always a sports score.

---

[2] LN, it seems, is extremely proud of the fact that their system is almost completely invulnerable to automated downloading. Now, if they would devote the same level of attention to correcting the bugs that results in LN searches differing by 10% when two identical searches are run within 10 minutes of each other...

[3] Yeah, right, just like the flexibility in the face of technological innovation shown by the music industry, video rental stores, mass market bookstores, classified advertising, and so forth. Okay, so they probably won't innovate, they will just die.

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 our now-30-year "Levant" data set primarily looks for stories containing the names or synonyms of the six actors we are tracking: 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 when they are working), but they do present problems when downloading—we originally did this using [gasp!] a phone modem [25]—or when one is paying by the story.[4]

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.

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 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:

[4] Payment-by-story is the current initial bargaining position used by the bulk providers, even when their antiquated search engines produce vast quantities of irrelevant stories. We have it on good authority, however, that if one has sufficient funding, they will eventually get more reasonable. This issue is irrelevant, of course, in real-time data acquisition strategies, where the marginal costs are close to zero but all of the filtering is done after the downloads.

- Exact duplicates, where a local source simply reprints the contents of an international newswire story. This is what newswires are for, so this 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. In recent developments at Penn State, 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. 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.

When used in a predictive mode, duplicates are not necessarily a bad thing, since they will generally amplify politically-relevant 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. Thus, some in-progess event data programs are beginning to count the number of times a specific story—based on both the type of event and the inferred location—was "duplicated" in other media sources in order to use this score as a proxy for the importance of the event.

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 fifteen 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—Reuters went through something close to an organizational near-death experience during the period 1998-2002 [21] and the frequency of its reporting dropped dramatically during that time, but this was followed by a near-exponential increase in reports in the mid-2000s, resulting in a statistically-problematic "dip" in

[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.

the story frequency. Idiosyncratic policies on updating, corrections and the broadcasting of summaries will also affect story frequency.

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, and the academic project NewsBlaster.[6] 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 SVM—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 [15, 16] and the Conflict and Peace Data Bank (COPDAB) developed by Edward Azar [3, 1, 2]. Both were created during the Cold War and 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 [17, 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

[6] Academic but not, alas, open source, and in all likelihood soon to be rendered irrelevant by Google News.

conflict-cooperation continuum that many researchers consider inappropriate. Although there have been efforts to create alternative coding systems—most notably Lengs Behavioral Correlates of War (BCOW) [14]—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. Established protocols for training and maintaining consistency among coders further constrained efforts to improve WEIS and COPDAB once these were institutionalized. As a consequence, endeavors such as Tomlinson's modification of WEIS [31] and the Global Event Data System (GEDS) project extensions of COPDAB [7] produced only marginal changes.

In contrast to human coding, automated coding 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. This elimination of questionable phrases itself represents an improvement in the coding system. 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. Consequently, in the mid-1990s, the Protocol for the Analysis of Nonviolent Direct Action (PANDA) [4] was developed in an initial experiment with the combination of automated coding and a new ontology focused on sub-state actors, followed by the development of the Integrated Data for Events Analysis (IDEA) [5] system, designed as a super-set of several existing ontologies along with innovations such as the use of tertiary (4-digit) event categories and codes for non-human events such as natural disasters.

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 [27]. 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. 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 2-digit cue categories 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. Consequently, we developed CAMEO, which is specifically designed to code events relevant to the mediation of

violent conflict but can also be used for studying other types of 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) [16] . 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 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: that is, codes that were dependent on *who* was engaged in the event, not just *what* was being done.

In addition, we made CAMEO consistent with respect to the numerical order of its main cue categories. Unlike WEIS and IDEA, we start with the most neu-

tral 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. Following the model of the IDEA codebook, the CAMEO codebook exists in both printed and web-based formats. We have also followed the lead of IDEA in introducing 4-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 settlement. We anticipate that the tertiary categories will be used only rarely, not be used but they are available if a researcher wants to examine some very specific behaviors that might be useful in defining patterns. The tertiary categories also clarify further the types of event forms included in the secondary and primary categories, leading to more precise and inclusive coding.

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, six-month revision of WEIS for a single NSF grant, we ended up spending almost three 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—that were developed for specific region of the world (initially the Balkans and West Africa) needed relatively little work to produce useable data with global coverage. This was partially unexpected but consistent with our earlier experiments in extending the data sets, which have always used a shared *.verbs* dictionary despite using specialized *.actors* dictionaries. For example, 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.

In the long run, it might be possible to re-define the entire CAMEO coding ontology using the standardized *WordNet* synsets, rather than using the current categories that were developed inductively. This would again help align the event coding with the larger NLP community, and probably simplify its use in languages other than English. We currently have, in house but not fully integrated, a completely re-organized *.verbs* dictionary[7] that combines the existing verbs according to the *WordNet* synsets, as well as adding additional *WordNet* synonyms not found in the original dictionaries.[8] The new dictionary also implements about two dozen com-

[7] The results of extensive work by Marsha Sowell

[8] Most of these are quite obscure and oftentimes archaic...*WordNet* at times feels like one is permanently in "Talk Like a Pirate" day...

mon noun synsets (for examples units of currency, measurement, and weapons). As soon as we get the resources to finalize and test this new dictionary, we will make it available, and this should increase the robustness of the coding as well as simplifying any future updates of the coding scheme.

In addition, a number of recent discussions within the policy community have suggested that the following categories of events might be useful additions

- Natural disasters such as hurricanes, drought and earthquakes; IDEA has already done this
- Epidemic disease; it would be particularly useful to standardize these against code used by international public health authorities;
- Economic and financial transactions – neither CAMEO nor IDEA is very good on these, and there is a lot of interest in them;
- Probably more detail on non-political criminal activity.

No list of codes will be extensive: for example according to *Time* (5 March 2012), one of the tactics used by the religious authorities in Iran against Ahmaddinajad's party is to accuse them of using witchcraft (or at least some Persian variant thereof). Which is probably not the sort of code one would use in coding European politics—though one must note that it occasionally does arise in the US, notably by the televangelist Pat Robertson, who perhaps not coincidentally also just endorsed the legalization of marijuana—but is also salient in Kenya, Nigeria and Indonesia. So this might, in fact, be a useful tertiary code.

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 analysts 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 non-violent popular revolutions in Eastern Europe or the more recent "Arab Spring."

Because WEIS and COPDAB were state-centered, they 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, WEIS and CAMEO ignored sub- and non-state actors.

A major breakthrough in the systematic coding of sub-state actors came with the PANDA project [4], 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 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.

The most recent iteration of TABARI implements a general actor-agent system so that instead of requiring explicit specification of terms such as "Israeli military" or "Afghan militants," any combination of the form $< actor >< agent >$ automatically generates the appropriate code.[9] Furthermore, we have implemented a nearly comprehensive dictionary of common nouns that could refer to agents using *WordNet*; this involves about 2,200 agents. This innovation both substantially reduces the size of the original *.actors* dictionaries, which contained a large number of phrases which varied only with the national identifier (e.g. "Israeli military", "Jordanian military", "Egyptian military") as well as increasing the overall generality of the system, as well as increasing the accuracy of the noun-phrase identification in the parsing and improving the speed of the program because of the reduced size of the *.actors* dictionaries.

Three principles underlie the CAMEO actor coding system. First, codes are composed of one or more three-character elements. Since the current system now can generate codes automatically, we have seen code strings as long as 24 characters, though the typical code consists one, two or three of these elements (and therefore three, six, or nine character codes). These code elements are classified into a number of broad categories, such as state actors, sub-state actor roles, regions, and ethnic groups.

[9] Explicit strings take precedence over generic actor-agent combinations so, for example, if one knew that the phrase "Tamal militants" always referred to the Tamal Tigers, one could specify the code LKAREBTTG rather than defaulting to a generic code.

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 [14] is another example of a hierarchical scheme in the event data literature.

Third, we utilize 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 [23] 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`). Ze'ev Maoz and Errol Henderson are currently working on a religious ethnic classification scheme under funding from the Templeton Foundation, and we may switch to this once it becomes available: we are interested only in the utility, not the proliferation, of coding schemes. Fortunately with fully automated coding, implementation of a new coding system involves only the labor involved in the modification of a set of dictionaries, plus a nearly costless automated recoding, so these changes are quite feasible.

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. Due to the efforts involved in manually downloading the news stories, the KEDS project had focused on a small number of geographical areas, primarily the Levant, with ten-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 two-year—time periods, and graduate student research by Ömür Yilmaz and Baris Kesgin had produced very detailed dictionaries for

Turkey, but that was it. Systems focusing on real-time coding, in contrast, generally are interested in coding the entire world, as there is little reason not to do this.

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. 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.

NER is part of the more general natural-language processing issue of "Named Entity Recognition and Resolution" systems for the recognition of names within text, and also resolving equivalent names, e.g. "President Obama," "President Barak Obama," "United States President Barak Hussein Obama" and so forth. Some of these methods are very sophisticated—for example using conditional random fields and hidden Markov models—though it is not entirely clear these are needed for NER when dealing with political actors found in news reports, who have fairly regularized names and titles.

Neither of these techniques scaled to large text corpuses, 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 Phase I corpus was limited. *ActorFilter*, unfortunately, had not been designed for a project of large magnitude and while it could be used on a sample, it slowed to an unusable crawl on massive files. Consequently, we experimented with three supplemental approaches.

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 developed for the first phrase of ICEWS had around 20,000 entries, compared to the 1,000 or so entries typical in earlier regionally-specific KEDS work.

Following this experimental work for the first phase of ICEWS, we augmented a reference file used in earlier NSF-funded work on the Militarized Interstate Disputes dataset [29] with information in the *CIA World Factbook* and `rulers.org` to a comprehensive list of state names, major cities, regions 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. In some initial experiments with a very large real-time text base, *CountryInfo* in combination with the *WordNet*-based *.agents* dictionaries have proven surprisingly robust and in fact

country-specific NER may be less important than we originally assumed, as the international news reports generally include specific information about the role of an individual (e.g. "German Chancellor Angela Merkel") that the specific name is redundant except for a small number of very conspicuous international figures such as the U.S. president (and possibly Angela Merkel. But not Fijian Commodore Frank Bainimarama). Further experimentation and quality checks will be needed to assess this situation.

In the meantime, however, *ActorFilter* has replaced with a new open-source `Python` program, `PoliNER`, which has a similar function but is 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.

5 Coding and Post-processing

5.1 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. The code seems quite robust in a Unix environment, and has now been cleaned up to remove all gcc warnings, and we have routinely run it on a variety of different systems.[10]

In our initial experiments with the ICEWS Phase I in the summer of 2008, coding approximately 26-million sentences 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 in those 26-million sentences (welcome to the world of large-scale text analysis: abandon all hope of control over the input).

In 2009, we gained access to a small, 14-processor cluster computer that was sitting unused (and undocumented) at the University of Kansas, which allowed us to begin implementing TABARI on a computer cluster, thereby dramatically increased coding speed. 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

[10] Though we have yet, despite long efforts, found anyone to convert it to the Windows system of partially-debugged device drivers, suggesting that in a large-scale text processing environment, Unix rules at least for research software. Lockheed Martin has developed Java implementation called JABARI-NLP for applied proposes which will run in Windows, and this was used in the later phases of ICEWS.

re-combined the output files at the end of the run.[11] 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 a 26-million story corpus to be coded in about six minutes. The initial set-up, of course, took quite a bit longer, but this was particularly useful for weeding out 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 splitting, so the gain is almost linear.

5.2 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.

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.

Furthermore, once geolocation has been implemented, as discussed in the next section, the "one-a-day" filtering can be replaced with location-based coding, which will allow the possibility of multiple events in distinct locations on the same day. This has been implemented in some experimental systems, and as expected is particularly useful in high-intensity conflict situations such as the current civil war in Syria.

5.3 Geolocation

A still missing component of the system is the ability to tag the location where the event occurred. Currently, event datasets report the country of origin of the actors, and from this researchers deduce where the event occurred. Thus, when utilizing

[11] We did this with customized software; the widely available "mapreduce" package for cluster computers does this more generically.

existing event data sets to analyze things like civil war, common practice is to aggregate events at the state level, as opposed to the sub-state district or municipal level. For large countries with numerous ongoing domestic disputes, like India, the lack of sub-state level geo-coding inhibits researchers' ability to model different regions separately. Despite the large number of open-source NLP programs (discussed in section 6.1), there is a relative dearth of software designed to geo-code event locations. Nevertheless, there are several automated systems for doing this type of tagging and we expect to see these fully implemented in the near future: the required modifications for TABARI to use that information appear to be minor.

The National Geospatial-Intelligence Agency maintains a continuously-updated database of approximately 5.5-million geographical place names (`http://earth-info.nga.mil/gns/html/`) called the GEOnet Names Server (GNS), and each place name is assigned specific latitude-longitude coordinates. The GNS apparently provides unique identifiers to locations which can be referred to by multiple spellings—for example in the transliteration of Arabic and Chinese names—which will further increase the accuracy of the data for analysis and visualization purposes. A large-scale, in-progress event data program is experimenting with an approach that would provide geo-location data in the form of latitude and longitude for events occurring in unambiguous locations. The main challenge is to empirically determine which place name should be assigned to the specific event, especially when multiple events and place names occur in a single article. Preliminary results suggest that this is feasible by empirically calculating which place name is nearest to the action in the text, though further fine tuning will likely be needed.

6 Open Issues

6.1 Open Source NLP[12]

Earlier coding software was largely self-contained. Now, however, it makes far more sense to leave the NLP software development to the computational linguists whenever possible, and focus only on those remaining tasks specifically required for coding events. This would be consistent with the broader incorporation of automated text processing into political science [20] in contexts such as the analysis of legislative debate and party platforms.

Major open-source NLP software sites include:

- Open-NLP `http://opennlp.apache.org/`
- GATE; `http://gate.ac.uk/`
- University of Illinois Cognitive Computation Group: `http://cogcomp.cs.illinois.edu/page/software`

[12] This section in particular has benefitted from a variety of discussions with David Van Brackle, Will Lowe, Steve Purpura and Steve Shellman

- Stanford NLP Group: `http://nlp.stanford.edu/software/index.shtml`

LingPipe's "Competition" page (`http://alias-i.com/lingpipe/web/competition.html`) lists—as of March 2012—no fewer than 23 academic/open-source NLP projects, and 122 commercial projects. This is quite different than the situation in statistical software, where at present there are only four major systems in wide use (SPSS, SAS, Stata and R), and perhaps a dozen or some additional specialized systems.

NLP tasks relevant to event coding include the following:

- Full-parsing. An assortment of full-parsers are available, and the *TreeBank* parse format (`http://www.cis.upenn.edu/~treebank/`) appears to be a fairly stable and standard output format, so a researcher could use the parser of his or her choice so long as these could produce *TreeBank*-formatted output.
- 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 and complexity. Parts-of-speech (POS) marking eliminates this problem. Stemming—most frequently the Porter stemming algorithm for English—further simplifies and generalizes coding dictionaries.
- Synonym and relational dictionaries. The *WordNet* lexical database provides a nearly comprehensive list of synonyms ("synsets"), hyponyms and hypernyms for the English language; this could be used to replace specific instances of nouns and verbs with general classes, and as noted above, we have fully implemented these in our *.agents* dictionary and have most of the basic work finished to implement in in the *.verbs* dictionary. Various projects have also assembled extensive lists of specialized words such as currencies, occupations, first names, titles and so forth: See for example the files in the `LbjNerTagger1.11.release/Data/KnownLists` folder that can be downloaded as part of the Illinois NER system at `http://cogcomp.cs.illinois.edu/page/download_view/4`.
- Regular expressions. Given the ubiquity of regular expressions in the contemporary computing environment—regular expressions have been called "the calculus of string processing"—it would be very useful to allow these to be used as the pattern-specification language, rather than the ad hoc syntax used in most of the existing systems.

The use of these tools would align automated event coding and machine-assisted coding with the research output of the larger NLP community, and as their tools improved, we could incorporate those improvements into our work immediately. Two systems have already moved in this direction. Lockheed modied their JABARI-NLP coder—originally just a proprietary Java version of the open-source, C/C++ TABARI —to produce JABARI-NLP, which uses open-source NLP software for considerable pre-processing of the texts. This resulted in substantial improvement in coding accuracy, particularly by insuring that the event target identified by the coder was actually syntactically part of the verb phrase generating the event.

More generally, off-loading most of the NLP tasks to sophisticated pre-processing programs means that programs for automated coding or machine-assisted coding

can be much shorter and in more easily maintained forms using a robust scripting language—probably, given its wide application in NLP tasks, `Python`—rather than in a fully compiled language such as `C/C++` or `Java`. Scripting languages take a performance hit but, due to their internal memory management, are usually considerably shorter and easier to maintain. In addition, since almost all high-volume coding will now be done in parallel environments, speed is less critical than when coding was done on single processors.

7 Near-Real-Time Coding

The widespread availability of real-time news reports on the Web and through alternative media such as RSS feeds opens the possibility of near-real-time coding of atomic events. Implementation simply involves linking together the appropriate web-scraping scripts, formatters, duplicate detection programs, an automated coder, a database such as `mySQL`, and a web-based interface.[13] We implemented an experimental system in 2008-2009, and our 18-month experiment has found at least three characteristics of the data that should be taken into account in the design of any future systems.

- While *in principle* one could get real-time coding—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 reports became available.
- These are definitely not "build and forget" systems due to the changing organization of the news web sites. We found Reuters went through three or four major reorganizations of their web site during the period we were coding data, and in one instance was off-line for close to a week. The changes in code resulting from these reorganizations were relatively minor, primarily dealing with the locations of files rather than the file formats, but it necessitated periodic—and unexpected—maintenance.
- A semi-automated NER system would be needed to track new actors entering and changing their status in the system, for example a politician losing office or a militant leader getting killed. However, with machine-assisted NER tools, this is within the capabilities of a relatively small project.

[13] UCDP provides a sophisticated model for this for its various composite event sets: `http://www.pcr.uu.se/research/datasets/generate_your_own_datasets`

7.1 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 write the equivalent of TABARI 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 modifications would be relatively easy; the linguistic theories of Noam Chomsky suggest that the required modifications will fall into a relatively small number of categories.

The second possibility would be to use NLP tools to handle the parsing but still use language-specific *.verbs* dictionaries. These could also contain new idiomatic phrases—which are likely to be quite important and quite unsystematic across language and cultures—although these new dictionaries would involve considerable work. That effort might, however, be justified in the cases of languages where there are a large set of news sources, particularly on local events, which do not overlap well with the international English-language media: Spanish and Arabic come to mind, as would Chinese if an independent press develops in the PRC.

The final possibility is to use machine translations into English of the source texts, and then continue to use the English-language coder and dictionaries to generate events. 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. These are often based on statistical methods intended simply to provide a recognizable sense of the text, not an eloquent rendition of it.[14] 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. To date, this appears to be a promising avenue for future development, but it is less clear whether it is functional at this point.

[14] Sometimes with unexpected results: as of 7 March 2012, Google Translate rendered the Norwegian phrase *Kaffebrennereit*—a ubiquitous chain of coffee shops in Oslo—into English as *Starbucks*, and helpfully provides *coffee burning shop* as an alternative translation.

7.2 Real-Time Coding

We used the 2009-2010 period to undertake an experiment in true real-time coding using RSS feeds. [15] 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 news 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 has 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

[15] This system was developed largely by Vladimir Petrov working at Kansas

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.

8 Conclusion

In a history of the first fifteen years of the KEDS/TABARI project [25], 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. In 1962, Deng Xiaoping famously (if politically prematurely...) 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 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,* has been translated into Chinese.[16] The cat, so to speak, is out of the bag.

The technological environment which provides the support for automated event coding continues to change rapidly. To date we still do not have an ideal environment for source texts: this would involve some sort of reasonably priced large scale database that could be explicitly used for data-mining, rather than the existing data services which were designed for "needle-in-a-haystack" searches using apparently quite antiquated software and which are still run by companies which regard all efforts at data mining as attempts to abscond with their [secondary] intellectual property.[17] That, however, is almost the *only* missing component at the moment: we clearly now have real-time news feeds, dictionary-based approaches that appear

[16] Or so I've been told, though I've not been able to locate this on the web. Itself interesting.

[17] Rather as the music industry regarded the advent of digital media as a criminal activity and embarked on the brilliant public relations strategy of suing their customers. We saw how that turned out.

to generalize to global coding despite their original applications to region-specific data, and a variety of open-source solutions that could be combined with existing software to provide improved parsing, geolocated data, and possibly applications to languages other than English.

Experimental and proprietary work with these new approaches is already underway, generally with very positive results, and we anticipate that we will see some major innovations in publicly-available large scale data sets in the very near future. To date, however, we still await the equivalent of *iTunes* for large-scale news text: the business model is almost trivial given that the marginal cost of producing such collections should be close to zero, but as the information revolution has shown repeatedly, plausible models can sit unimplemented for years before finally overturning long-established businesses in a matter of months. Change will almost certainly happen: this is a matter of "when", not "if." Be patient. But not too patient.

Acknowledgements This paper was presented at the conference New Directions in Analyzing Text as Data, Harvard University, 5-6 October 2012. This research was supported in part by 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`).

References

1. Azar, E.E.: The conflict and peace data bank (COPDAB) project. Journal of Conflict Resolution **24**, 143–152 (1980)
2. Azar, E.E.: The Codebook of the Conflict and Peace Data Bank (COPDAB). Center for International Development, University of Maryland, College Park, MD (1982)
3. Azar, E.E., Sloan, T.: Dimensions of Interaction. University Center for International Studies, University of Pittsburgh, Pittsburgh (1975)
4. Bond, D., Bennett, B., Vogele, W.: Data development and interaction events analysis using keds/panda: an interim report (1994). Paper presented at the International Studies Association, Washington
5. Bond, D., Bond, J., Oh, C., Jenkins, J.C., Taylor, C.L.: Integrated data for events analysis (IDEA): An event typology for automated events data development. Journal of Peace Research **40**(6), 733–745 (2003)
6. Chenoweth, E., Dugan, L.: Rethinking counterterrorism: Evidence from israe (2012). Working Paper, Wesleyan University
7. Davies, J.L., McDaniel, C.K.: The global event-data system. In: R.L. Merritt, R.G. Muncaster, D.A. Zinnes (eds.) International Event-Data Developments: DDIR Phase II. University of Michigan Press, Ann Arbor (1993)
8. Dugan, L., Chenoweth, E.: Moving beyond deterrence: The effectiveness of raising the expected utility of abstaining from terrorism in israel (2012). Working Paper, University of Maryland
9. Gerner, D.J., Schrodt, P.A., Francisco, R.A., Weddle, J.L.: The machine coding of events from regional and international sources. International Studies Quarterly **38**, 91–119 (1994)
10. Gleditsch, N.P.: Special issue: Event data. International Interactions **38**(5) (2012)
11. Howell, L.D.: A comparative study of the WEIS and COPDAB data sets. International Studies Quarterly **27**, 149–159 (1983)
12. Kahneman, D.: Thinking Fast and Slow. Farrar, Straus and Giroux, New York (2011)

13. King, G., Lowe, W.: 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)
14. Leng, R.J.: Behavioral Correlates of War, 1816-1975. (ICPSR 8606). Inter-University Consortium for Political and Social Research, Ann Arbor (1987)
15. McClelland, C.A.: World-event-interaction-survey: A research project on the theory and measurement of international interaction and transaction (1967). University of Southern California
16. McClelland, C.A.: World Event/Interaction Survey Codebook (ICPSR 5211). Inter-University Consortium for Political and Social Research, Ann Arbor (1976)
17. McClelland, C.A.: Let the user beware. International Studies Quarterly **27**(2), 169–177 (1983)
18. Merritt, R.L., Muncaster, R.G., Zinnes, D.A. (eds.): International Event Data Developments: DDIR Phase II. University of Michigan Press, Ann Arbor (1993)
19. Mikhaylov, S., Laver, M., Benoit, K.: Coder reliability and misclassification in the human coding of party manifestos. Political Analysis **20**(1), 78–91 (2012)
20. Monroe, B., Schrodt, P.A.: Editors' introduction: The statistical analysis of political text. Political Analysis **16**(4), 351–355 (2008)
21. Mooney, B., Simpson, B.: Breaking News: How the Wheels Came off at Reuters. Capstone, Mankato, MN (2003)
22. O'Brien, S.P.: Crisis early warning and decision support: Contemporary approaches and thoughts on future research. International Studies Review **12**(1), 87–104 (2010)
23. Russett, B.M., Singer, J.D., Small, M.: National political units in the twentieth century: A standardized list. American Political Science Review **62**(3), 932–951 (1968)
24. Schrodt, P.A.: Statistical characteristics of events data. International Interactions **20**(1-2), 35–53 (1994)
25. Schrodt, P.A.: Twenty years of the Kansas event data system project. The Political Methodologist **14**(1), 2–8 (2006)
26. Schrodt, P.A.: Precedents, progress and prospects in political event data. International Interactions **38**(5), forthcoming (2012)
27. Schrodt, P.A., Gerner, D.J.: Validity assessment of a machine-coded event data set for the Middle East, 1982-1992. American Journal of Political Science **38**, 825–854 (1994)
28. Schrodt, P.A., Gerner, D.J., Yilmaz, Ö.: Conflict and mediation event observations (CAMEO): An event data framework for a post Cold War world. In: J. Bercovitch, S. Gartner (eds.) International Conflict Mediation: New Approaches and Findings. Routledge, New York (2009)
29. Schrodt, P.A., Palmer, G., Hatipoglu, M.E.: Automated detection of reports of militarized interstate disputes using the SVM document classification algorithm (2008). Paper presented at American Political Science Association
30. Tetlock, P.E.: Expert Political Judgment: How Good Is It? How Can We Know? Princeton University Press, Princeton, NJ (2005)
31. Tomlinson, R.G.: World event/interaction survey (WEIS) coding manual (1993). Mimeo, Department of Political Science, United States Naval Academy, Annapolis, MD