Analyzing International Event Data: A Handbook of Computer-Based Techniques

Philip A. Schrodt
and
Deborah J. Gerner

PDF:
http://eventdata.psu.edu/papers.dir/automated.html

Version 1.1
October 2000 with revisions in March 2012
Last update: March 23, 2012

Provenance:
This was originally part of a book-length manuscript which, like everything in the KEDS project, was never published due to the lack of a commercially-viable audience. The first three chapters continue to be used, and we have now reformatted these in a stand-alone format which includes the relevant bibliographic citations, as well as converting—with considerable assistance from Krista Lombardo and support from National Science Foundation grant SES-1004414—to LaTeX. In the process, I have done some updating of the text, though this is by no means comprehensive. Readers interested in the current state-of-the-art should consult various review papers at http://eventdata.psu.edu/papers.dir/automated.html.

Chapter 2: Fundamentals of Machine Coding

1 Introduction

As noted in Chapter 1, one of the most notable changes in event data analysis that occurred during the 1990s was the shift from human coding to fully automated machine coding of events. The machine coding systems do not completely eliminate the role of humans in coding, since the existing systems still require the systematic development of dictionaries that are used by the specialized coding programs to convert textual reports into event data. But machine coding shifts the focus of those efforts from the repetitious coding of routine events to the problem of determining general language patterns that correspond to events. And once the dictionaries have been developed, the coding itself can be done in a few hours. This flexibility has led to a renewed production of event data and to greater experimentation with event coding schemes.

The bulk of human interactions utilize, and are recorded in, natural language. But to date most systematic studies of human behavior that follow the scientific model of research have analyzed numbers. In some fields of study—most notably economics and geography—this disjuncture is only mildly inconvenient, for most of the behaviors of interest can be represented numerically with little distortion. But in other fields—for example, anthropology, communications studies, psychology, political science, and sociology—considerable information is lost in the transition from a natural language representation of behavior to a numerical representation. Furthermore, because those transitions are usually mediated through human coders who interpret the information through a haze of preconceptions, misunderstandings, personal biases, and sloth, with the result that substantial unsystematic error is introduced into the data.

The World Wide Web and the personal computer revolution have made available in machine-readable form a vast amount of information about social behavior instantiated as natural language texts. These range from routine news reports to political speeches, debates, and policy statements to the manifestos of millenarian UFO cults. In addition to Web-based sources, many other traditional materials such as transcripts, interviews, protocols, and ethnographies are either available in electronic form or readily converted to that form.

Until recently, social scientists had only a limited number of tools to systematically locate and analyze this information; this stands in contrast to the massive efforts devoted to developing software for the analysis of numerical information. Computational tools employing simple pattern recognition—for example, keyword-in-context (KWIC) indices, Boolean search engines, and a variety of systems for tabulating words and phrases—were available from almost the beginning of computational social science, but few of these systems made even the most basic linguistic distinctions such as differentiating between subjects and objects in sentences. In retrospect, the computational power required for the sophisticated analysis of textual material simply was not available.

During the 1980s and 1990s, there was an exponential expansion in the capacity of personal computers. This was accompanied by substantial work in computer science and linguists on natural language processing (NLP), and computer programs are now capable of performing many tasks once assumed to require human attention. Grammar checking and document summary, for example, is now a routine feature of many word-processing programs; automated indexing and document retrieval are widely used in specialized applications. Inexpensive language translation programs—a task abandoned as impossible in the 1950s and 1960s—are now available to handle translations of technical documentation, web pages, and business correspondence. High capacity computers can accommodate the large numbers of grammatical rules, idiomatic expressions, and context-sensitive substitutions that are necessary to produce credible translations where earlier systems, constrained

to simpler rules and word-for-word dictionary translations, had failed.

None of these systems is perfect. Machine translations are usually inferior to those of a fluent translator and break down completely when dealing with oddly constructed or excessively complex sentences as might be found in poetry, classical literature, or even political rhetoric. But machine translation is typically superior to that provided by a translator with limited fluency, and when translating routine material such as maintenance manuals, a customized translation program with a broad range of specialized vocabulary may actually produce a document superior to that provided by a human without the domain-specific knowledge.

This change in the ability of machines to handle NLP problems was reflected in event data research: in 1990 almost all event data projects used human coders, whereas in 2000 almost all projects used automated coding. This transformation meant that projects that once would have required tens of thousands of dollars, a flock of student coders with a complex supervisory infrastructure, and months of painstaking effort could be done by a single researcher in a few days or weeks, provided appropriate pre-existing dictionaries were available. This development has, in turn, led to a series of less-obvious spin-offs such as the ability to easily experiment with new event coding systems such as PANDA and with the coding of internal actors [15, 12].

The KEDS machine-coding program was the first machine-coding system available to the academic community that could produce research-quality event data—that is, data accepted in top-ranked publications in the political science discipline.[1] But KEDS is neither particularly innovative

[1]This includes the *American Political Science Review* [12], *American Journal of Political Science* [31], *International Studies Quarterly* [10], and *Journal of Conflict Resolution* [32, 7].

We have been asked repeatedly whether superior automated coding systems exist in the classified sectors of the U.S. government. We have not done classified work, and therefore are in no position to answer this definitively; and of course if we *had* done classified work, we couldn't answer the question at all. But since the issue is raised continually, let us speculate.

DARPA has sponsored extensive work on NLP at academic institutions, some of which was specifically oriented toward extracting information from news stories. The MUC Project [9, 3] was the most conspicuous of these efforts and involved a number of institutions and projects; the substantive domain was newswire descriptions of terrorist incidents. A great deal of additional work in computational linguistics was done during the 1980s and 1990s that could have been applied to the machine-coding problem, although—as with 1990s developments in cryptography—there is little evidence that classified work in NLP substantially outpaced work in the commercial sector.

The likelihood of substantial investment in classified machine-coding work is further reduced by the limited interest that the policy community showed in event data from 1985 to 1995. When interest revived in the mid-1990s in the context of early warning, the policy community was initially willing to put up with the considerable expense and delay of human-coded data (e.g. the State Failure Project's use of GEDS), or relied on the VRA coder—a project adapted from academic work and itself initially subject to substantial delays in development—for machine-coded data. Even KEDS, despite its simplicity, received attention, and under government contracts KEDS dictionaries were developed to code events in a number of crisis areas.

None of these actions suggest the existence of a readily available but secret coding system, nor has any conversation we have had with individuals involved in policy-oriented projects for the U.S. government, European governments, or U.N. agencies. And we've had a lot of conversations, in one instance suggesting to someone at a U.S. government installation in Hawai'i that while we appreciated their interest in KEDS , our understanding was that they were already using VRA-Coder, and received an appreciative phone call fifteen minutes later that wow! the guy using it is just down the hall! Your tax dollars at work. Anecdotal reports have suggested the absence of such as system: The usual suspects for developing such a system such as IBM, BBN and SAIC had substantial NLP expertise in fields such as automated filtering, indexing, and abstracting but had nothing available for event data coding per se.

This does not completely eliminate the possibility of an advanced event coding system located somewhere in the defense or intelligence communities—housed, perhaps, in the same facility that preserves the bodies of the aliens recovered from the UFO crash in Roswell, New Mexico—but if such a system exists, it is very well hidden and is unavailable even for an analysis such as the White House-mandated State Failures Project.

March 2012: While I remain very skeptical about the existence of a *secret* system, the DARPA-funded Integrated Conflict Early Warning System Project (ICEWS; [26]) made substantial investments in automated coding technologies. As a result of the ICEWS project, several additional proprietary coders have either been developed or are under development: these include Lockheed ATL's JABARI family and Strategic Analysis Enterprise's XENOPHON,

nor original. It is instead simply a user-friendly implementation of a number of general NLP principles that were common in computer science and computational linguistics by the late 1980s.

In comparison to many other NLP problems, the machine coding of event data is a relatively simple task because most event categories are defined by sets of transitive verbs. In event coding, the subject of the sentence is the *source* of the event, the verb determines the *event code*, and the object of the verb is the *target*. (Many discussions of event data use the word *actor* to refer to the source of the event. In our discussion, "actor" refers to the set of entities that can be sources or targets.) KEDS achieves nearly perfect coding when presented with a declarative English sentence such as

> Saudi Prince Alwaleed bin Talal, one of the world's richest investors, will donate $30 million to rebuild the two Lebanese power stations destroyed by Israeli attacks, an official source said on Friday.

The program can also deal with a number of standard grammatical constructs such as compound subject, objects and sentences, pronouns, and passive voice. Thus, KEDS is successful on the sentence structures most commonly used to describe events. But the program has limits as well: a sentence such as the following will be either coded incorrectly or not coded at all:

> Two Israeli civilians were killed and a third was seriously wounded in northern Israel on Thursday when Katyusha rockets fired from Lebanon struck the town of Kiryat Shmona, the Israeli army said.

Among the elements of this sentence that would confuse KEDS is the fact that the subject of the sentence (Israel) is the object of the attack, the unusual "...killed and a third was ..." construction of a compound phrase, and the fact that the verb phrase that best defines the political event—"rockets ...struck ...town"—is near the end of the sentence. This lead sentence also does not identify who fired the rockets—KEDS could only note they came from Lebanon—though later sentences in the story identify the Hizbollah guerrilla movement as the presumed source. A number of general examples of the KEDS approach, and of Reuters more generally, will be discussed later in this chapter.

As noted in the preface, the discussion in this chapter should be taken as how the design of an automated coding system should begin, not necessarily where it should end. Because of the close connection between event data and the subject-verb-object structure of a sentence, most of the core features of KEDS will be found in any machine-coding system designed to code event data. However, more sophisticated capabilities—for example, comprehensive assignment of parts-of-speech (including resolution of ambiguous cases where appropriate), the parsing of subordinate clauses, syntactic identification of unknown words, and a larger repertoire of sentence forms—could easily improve on the performance of KEDS, and would be well within the capabilities of contemporary hardware and software. These extensions will be discussed at the appropriate points in this chapter. However, just as every pilot at the controls of a 747 first flew solo in a single-engine aircraft, there is much to be said for mastering the simple before introducing the complex, and that approach will be used in this chapter.

The discussion throughout this chapter will use Reuters lead sentences as examples, and the WEIS coding system as the reference event coding scheme. Nonetheless, most of the discussion

`http://strategicanalysisenterprises.com/services.php` as well as the inclusion of event coders on top of existing software, for example the "Behavior and Events from News" (BEN) system built into Social Science Automation's *Profiler Plus* and efforts by BBN and IBM to use their existing NLP "event triple"—subject-verb-predicate extraction—software for event coding.

should generalize to any news source using the traditional "pyramid style" of constructing a report, and to any event coding scheme that focuses on the subject-verb-object (SVO) structure of a sentence. In the discussion of WEIS, the phrase "cue category" refers to the general, two-digit categories such as 020 "Comment," 090 "Request," 170 "Threaten," and 220 "Force."

2 A [Brief] Defense of Human Coding

While most of this chapter, and this volume, emphasizes the advantages of machine-coded event data, some cautionary notes are in order concerning circumstances where human coding is more appropriate than automated coding. In essence, machine coding is the preferred approach for the production of a very large number of events from a set of texts that are reasonably standard in form and content. This characterizes most event data projects, but it does not apply to all circumstances where text might be coded for political analyses. Human coding may be preferable when some combination of the circumstances listed in the first column of Table 1 apply to a project.

Table 1: Comparative Advantages of Human versus Machine Coding

Advantage to Human Coding	**Advantage to Machine Coding**
•Small data sets	•Large data sets
•Data coded only one time	•Data coded over a period of time
•Existing dictionaries cannot be modified	•Existing dictionaries can be modified
•Complex sentence structures	•Simple sentence structures
•Metaphorical, idiomatic, or time-dependent text	•Literal, present-tense text
•Paragraph or story as the coding unit	•Sentence as the coding unit

Small Data Sets
Machine coding is appropriate for the *large-scale* production of data. If a project is sufficiently limited that the entire data set can be coded by a small number of competent and trusted human coders, then use human coding. The ability of any native speaker of a language to correctly parse and interpret a set of sentences, irrespective of their complexity, far exceeds the abilities of the most sophisticated computer programs, and this is a tremendous potential resource.

One-Time-Only Coding
As we will note below, one of the greatest advantages to machine coding is its stability over time. Many of the problems of human coding become relevant only when a project is done over an extended period of time, as issues of coder motivation, training, and turnover begin to overweigh the advantages of human natural-language comprehension.

Coding System Requires New Dictionaries
Developing the dictionaries required for machine coding can involve a substantial amount of effort: the original KEDS and PANDA dictionaries took about four person-years of effort to develop. Most academic researchers using machine coding have been willing to share their dictionaries, and most of the effort of dictionary development can be bypassed if a research project can employ a coding scheme similar to one previously developed for machine coding. (Typically the noun phrases used to code actors need to be modified, but relatively little modification is required for the verb phrases that determine the events.) However, if the research requires a coding system radically different

from any previous effort—and most KEDS-based efforts have focused on coding systems similar to WEIS—the labor involved in dictionary development may not compensate for the advantages of machine coding.

The Text Source is Grammatically Irregular

Most news reports are relatively simple in their grammatical structure, and relatively predictable in their content. News sources report "who, what, when, where and why" and usually do this in reasonably short, well-structured sentences. This is particularly true of lead sentences—the first sentence of an article that summarizes its content.

But not all political-relevant texts have this character. Counter-examples include direct quotations (even within news articles), speeches, political manifestos such as party platforms, treaties, and legal documents, and historical narratives that are interspersed with interpretive commentary. Sentences with a complex grammatical structure—for example, the use of multiple subordinate clauses, or wide separation between a verb and its object—are likely to be incorrectly coded, particularly by a simple parser such as that used in KEDS.

The Text Source Contains Extensive Idiomatic and Metaphorical References, or References to the Past or Future

Machine coding works best when most sentences can be assessed according to the most common dictionary definition of their words. Sentences containing elaborate idiomatic or metaphorical references—notoriously, the use of military terminology such as "blasts," "attacks," "devastated," and "destroyed" to refer to *verbal* confrontations (or in descriptions of sporting events)—will be miscoded by a machine, but can usually be correctly interpreted by an informed human coder. Political rhetoric employs these techniques frequently. While short idiomatic expressions can be easily incorporated into a dictionary, these expressions can sometimes cause coding errors when the words are used outside of an idiomatic context, and idiomatic expressions can go in and out of vogue relatively quickly.[2]

(A good example of a text source that is very difficult to code by machine is the Foreign Broadcast Information Service, which mixes short, literal reports of official functions with more lengthy transcripts of editorial commentary and official position statements. Sentences in the latter sources are frequently both grammatically complex and metaphorical.)

The sparse-parsing approach used by KEDS deals poorly with the issue of time—events occurring in the distant past or future. Human coders, in contrast, deal with this very easily and rarely mistake, for example, a reference to the anniversary of an event for the reoccurrence of the event itself. More sophisticated parsers that may be available in the near future are less likely to have this problem, at least at the level of individual sentences.

The Coding System Relies on a Unit Larger Than a Sentence

[2]Some features of language seem to persist indefinitely: skimming the earliest Reuters leads available on NEXIS produced no obvious examples of anachronistic language but did provide the following two leads from 17 April 1979 which, except for the proper names, could as easily be found, as this is being written in early 2000:

> Israeli Foreign Minister Moshe Dayan has said Israel may have to decide whether returning the occupied Golan Heights is a price worth paying for peace with Syria.
>
> Israeli Defence Minister Ezer Weizman met the commander of the Christian forces in South Lebanon today and told him that Israel had agreed that a limited number of Lebanese government troops should be stationed south of the Litani River.

Human coding will be required if the coding system involves a large number of "judgement calls" that interpret the words of a sentence in some larger context. If there are noun phrases or verb phrases that would receive different codes depending on their context—for example if the code depends on text that precedes or follows the sentences—then machine coding is not appropriate. Under these circumstances, of course, maintaining consistent human coding may also be difficult.

With all of these caveats in mind, the remainder of this chapter will make a strong case for the merits of machine coding for the production of large-scale event data sets from newspaper and newswire sources. In our experience, the advantages of machine coding clearly outweigh its disadvantages. In some applications, machine coding may also be effective even though the texts are not ideal. For example, one of our graduate students once used KEDS to code paragraph-level texts from Radio Free Europe transcripts—an application we would *not* have recommended had the student consulted us—and produced quite useable data. But machine coding is not the most effective, or most efficient, approach in all applications.

3 Justifications for the Use of Machine Coding

When we first began to develop machine coding, we viewed it as an inexpensive (and, given the funding environment, essential) but qualitatively inferior alternative to the human coding of event data. Ten years of experience have convinced us that it is in fact a superior approach. Rather than asking whether a machine-coding system comes sufficiently close to the efforts of human coders, the appropriate question should be whether the expense, inconsistency, and irreproducibility of human coding are justified. Our machine coding manifesto focuses on the following observations:

3.1 *No Published Human-Coded Event Data Project Has Ever Achieved Real-Time Coding*

In theory—and with nearly unlimited amounts of funding—it should be easy to develop a system for coding event data with human coders. Hire a bunch of coders, train them extensively until an appropriate level of inter-coder reliability is achieved, distribute the news reports, and start the coders coding.

In practice, the task is considerably more difficult, because event coding is one of the most boring tasks imaginable, yet it requires a great deal of specialized expertise about politics and geography. A coder must recognize the names of hundreds of political actors and their associated codes, and consistently retain the distinctions between thousands of verb-object combinations (e.g. "promise to send assistance" versus "promise to send troops"), and do this in a manner consistent with the interpretations of many other coders from a variety of backgrounds, levels of knowledge, and levels of motivation. The coders must do this, collectively, hour after hour, day after day.

They can't. Coding the first couple hundred events looks easy; in fact it is almost fun (and most principal investigators never get beyond coding a few hundred events). Coding the first couple thousand gets pretty boring. After that, the process is mentally painful. Philip Stone, the creator of *General Inquirer*, the first automated content program, recently observed, "Coding open-ended questions will reduce even the most deconstructionist English major to begging for a computer."[3]

So coders begin to move through the texts more quickly, ignoring the full set of events in compound sentences or those involving compound actors. The coders fail to show up for work, or when they do, they pay more attention to each other than to the texts.

[3]Workshop on Automated Content Analysis, 5th Social Science Methodology Conference, Cologne, Germany, 7 October 2000.

As new coders come in to replace those who have decided that even telemarketing is a preferable way to earn pizza money, the whole process of training and inter-coder validation has to be repeated. At this point the supervisors discover that several of the event definitions have "drifted" substantially in terms of how the [remaining] coders are actually implementing the protocol. Discussions of these issues reveal that experience with the larger collection of texts has shown the need for additional categories and some serious ambiguities in the original coding protocol. All of the texts that might be affected by these issues events must then be reviewed and possibly recoded. Productivity drops to a fourth or an eighth of that achieved in the initial training phases. The project drops further and further behind. This is the story of every human coding project with which we are familiar.

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 protocol changes and/or slacker-coders and so forth, usually ends up coding about six events per hour. Individual coders, particularly working for short periods of time, and definitely if they are a principal investigator trying to assess coding speed, can 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 6 events per hour is a pretty good rule of thumb.

Following the advent of low-cost computing facilities and easily constructed graphical user interfaces, machine-assisted coding has been proposed as the solution to the slow rate of human coding. While these methods almost certainly make coding less tedious, and probably result in higher levels of consistency, particularly in preventing typographical errors in the construction of codes, they appear to merely delay, rather than remove, the onset of fatigue, and do nothing to address the issue of inter-coder reliability in the interpretation of texts. Consequently the overall increases in coding speed have been modest. Once the human is in the loop, there is no quick technical fix and no, this time *won't* be different, however much you wish otherwise. The labor requirements of a string quartet have changed little over time, nor will the labor requirements of human event coding.

In the unclassified literature, there are no examples where an event data project succeeded in generating data at a rate even remotely close to real-time, and many projects have tried. There are reports—which may or may not be apocryphal—of WEIS's creator, Charles McClelland, maintaining that data set in retirement while sitting on his porch in Santa Barbara coding the *Los Angeles Times* every morning over coffee. But this model cannot be scaled to a multiple-coder project handling 1,000 new Reuters reports arriving each day, while concurrently generating a set of comparable data from past reports. Many projects—including some very well-funded projects—have attempted to attain real-time coverage, but to our knowledge none have succeeded. Many fail to generate any publishable data at all.

1,000 events per day?—one project circa 2011 reported an intake of 100,000 stories per day. Even assuming that 80% of these are duplicates or irrelevant—roughly the proportion of discards in the original ICEWS downloads from Lexis-Nexis—and could be removed completely efficiently, the remaining coding would require a team of around 400 coders assuming the realistic rate of 6 events per hour. This 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 any conceivable academic enterprise. 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—whether dedicated systems or distributed personal computing environments such as `XGrid`—and thus presents no such problems.

An extensive recent body of psychological work—see [5] for a popular treatment—indicates that the sustained decision-making required for human coding presents almost a perfect storm for

inducing fatigue, inattention, and a tendency to use heuristic shortcuts.[4] These physiological costs are far more deeply rooted than previously assumed, and can only be reduced, not eliminated, by improved coding protocols, training, coder selection and supervision. The human brain was simply never intended for the tasks we impose on coders.

3.2 *In Time Series Analysis, Stability is More Important Than Accuracy*

If Noam Chomsky is correct (about linguistics, if not politics), human coders come equipped with a magnificent asset that assists them in coding: a brain hard-wired to decipher language. No computer system currently used for event data coding even begins to match the ability of a human to correctly parse a complex sentence.

Unfortunately, human coders also come equipped with a large memory that includes a variety of background experiences, culture, prejudices, preconceptions, theories of political behavior, hindsight biases and levels of basic knowledge. The variance in these memories—and the profound effects they can have on the interpretation of a text describing political events—wreaks havoc on intercoder reliability. While two individuals might be taught to code in an identical fashion when dealing with abstract descriptions of political events, numerous psychological and perceptual biases affect the coding of actual events such as the Gulf War or the Arab-Israeli conflict. For example, if a coder has a preconception that Israel is allied with the United States and antagonistic toward Syria, the coder will tend to miss situations of conflict between Israel and the U.S. and situations of cooperation between Israel and Syria. Because these preconceptions differ between coders and vary with specific political contexts, their effects on coding are difficult to control.

In the abstract, the exercise of coder judgment is intuitively appealing. However, the assumption that the contextual knowledge of human coders will automatically improve the quality of event data ignores virtually everything we know about human cognition and perception. Jervis [16], Lebow [21], Vertzberger [37], Khong [18], and many of the articles in Hudson [14] and Singer and Hudson [33] describe the psychological distortions that occur in the human analysis of political events. Likewise, systematic evidence that humans—including experts in critical fields such as medicine and finance—vastly over-estimate their accuracy on complex tasks [34, 17]. It stretches credibility to assume these psychological factors will not also have an effect on human event data coders. Humans, unlike machines, read text amid a background of biases, expectations, and prejudices. In fact, the goal of an event data coding project should be to *remove* the element of human "judgment" and "knowledge of context" in determining event codes; these are as likely to increase the amount of noise in the data as they are to decrease it.

In the DDIR project, the most commonly recited story about the effects of coder biases was of a case in an early event coding project where the codes assigned by a female coder from a Third World country were deemed "unreliable" when compared to those of male, United States coders ([2]:224). We suspect that one would find equally large differences in white male upper-middle-class coders working over time. The coding of both WEIS and COPDAB spanned nearly a human generation, the initial coders of the 1960s growing up with "Leave It to Beaver" and the final cohorts in the early 1990s growing up with "Beavis and Butthead."

These preconceptions can cause problems even for experienced analysts. Laurance reports an experiment where

> in-house military analysts were used to help create a set of country scores for the military capability (naval in this case) of various [less-developed countries]. When we validated these scores with a set of outside experts, most were surprised that the PRC had scored so

[4] As well as leading to the consumption of excessive carbohydrates and presumably resulting obesity.

> high. Upon reflection we realized that the high scores were a function of the expectation that they would be very capable, since at the time they were in the 'communist enemy' category. In addition, the PRC was high on the collection and analysis priorities list of the intelligence community. More information existed on the PRC navy, compared to say Malaysia, Singapore, and India. [20]

More recently, a government agency undertook some extensive statistical tests of a human-coded data set to assess its utility for early warning purposes.[5] Statistical analysis pointed to sharp breaks at the change of the calendar year in many of the dyads, even when no obvious changes in the political situation were taking place. Further investigation showed that the coding project had assigned coders by dyad-year—individuals were given a full year of data to code, on the assumption that they would become familiar with the political actors of the period and code those consistently. This was a plausible approach from the perspective of coder training and the short-term consistency of the coding, but this protocol resulted in statistically-distinct breaks in the data at the time points where the coders were changed.

Event data are a time series, and consequently the *stability* of coding over time is critical. Stability is very difficult to maintain in human-coded projects, particularly because inter-coder reliability is nearly impossible to assess across cohorts of coders who are separated by as much as two decades in time and by a continent in space and culture. With appropriate record-keeping, one can easily assess how the current cohort of coders would code the events from the past, but it is impossible to determine how *past* coders would have coded *current* events. If a data set extends across a period of major political change—for example, the end of the Cold War in superpower relations, or the Oslo peace agreements in the Middle East—this can become quite problematic. In addition, the training protocols used in a project often are not properly archived, and the issues of intercoder reliability over time become dependent on tribal lore and fallible memories.

3.3 *3. Statistical Analysis is Designed to Detect Signals in Noisy Data*

Event data—whether human- or machine-coded—contain errors. As discussed in Chapter 3, coding errors are only one source of error in event data, and not necessarily the largest source. The text being coded is already an incomplete and biased record of the underlying events; the event coding scheme may be incorrectly aggregating some categories of events; and the statistical models in which the data will eventually be used in capture only some of the possible forms of the political relationships. Even if the assignment of actor and event codes was somehow perfect, the path from the political activity on the ground to the news reports being coded is a noisy process. Successfully identifying relationships amid that noise comes at the *analytical* stage of the project—both the development of the coding scheme and the statistical analysis—rather than in the coding stages.

The KEDS program generally assigns about 70%-90% of the same codes that a human coder would assign when working with the lead sentences of Reuters reports. In an experiment where dictionaries were optimized for the coding of a single day, PANDA achieved a 92% machine-coding accuracy; this probably represents the upper limit of accuracy for Reuters leads and a program using KEDS's sparse parsing approach [6]. In an independent test with a totally different type of source—the *Fortnight* chronology of events in Northern Ireland—Thomas [35] found that KEDS had an 82% accuracy on edited text and 73% accuracy on unedited text. While these assessments may be somewhat biased toward coding interpretations that favor machine-codeable events, KEDS-coded WEIS data sets also compare quite well with independently produced, human-coded WEIS data, as we will discuss in Section 4.

[5] We obtained this story first-hand, but the agency in question does not wish to be quoted.

This level of reliability between coder and machine is comparable to that found among human coders in event data projects: Burgess and Lawton ([8]:58) report a mean intercoder reliability of 82% for eight projects where that statistic is known. Because these reliability studies were done over a short period of time, in all likelihood they substantially over-estimate the true coding reliability within a data set such as WEIS or COPDAB that was coded by numerous different teams of coders across a number of years. We suspect, for example, that Reagan-era coders at the U.S. Naval Academy in Maryland interpreted at least some events differently than Vietnam-era coders at the University of Southern California, yet the WEIS series contains information coded by both groups.

But the human coding accuracy in some of those tests is quite low: King and Lowe's **(author?)** [19] coder 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.

The King and Lowe results are not anomalous: [24] find similar results in cross-checking the human coding of the Comparative Manifestos Project, a topic classification task quite similar to event data coding. Despite CMP's claim of 80% reliability, their experiments find the intercoder reliability averages about half that figure, with ranges quite similar to those found in King and Lowe; [28] report an agreement rate of 16% to 73% in another human event coding exercise. Those levels are almost certainly well within the range of existing machine coding technology, at least for atomic event coding.

4 A Comparison of Human- Coded and Machine-Coded Data

While the KEDS data discussed in Chapter 1 were shown to have considerable face validity, an equally important issue is whether they are comparable to human-coded WEIS data. In order to test this, we obtained human-coded data from the WEIS data set maintained by Rodney Tomlinson at the U.S. Naval Academy [36]. To compare the two data sets, the WEIS data were aggregated to monthly intervals and then compared with the monthly aggregations in a data set we generated with KEDS for the Middle East.

In this section, "WEIS" refers to the data set maintained by Tomlinson and "KEDS" refers to the data set that we generated; as noted earlier, the KEDS data are coded using the WEIS coding scheme. The WEIS data have been collected by Charles McClelland and several of his students. The collection of the data for 1982-1986 was supervised by Richard Beal and Frederick Roth at the National Security Council in the White House using *The New York Times* as a source; the data for 1987-1989 were coded by McClelland from the *Los Angeles Times*; and the data for 1989-1991 were coded under the supervision of Tomlinson using *The New York Times*. The WEIS set does not directly code for the Palestinians as an actor, but codes for an "ARAB COM" ("Arab community") that in the context of the dyads we are studying usually refers to Palestinians. This was converted to the PAL code in the comparison.

Four variables were correlated for each dyad: the total number of events, the net cooperation measured by the Goldstein scale, the number of cooperative events (WEIS codes 01 through 10, not including comments) and the number of conflictual events (WEIS codes 11 through 22). Table 2 summarizes these correlations.[6]

Three general patterns are evident from this comparison. First, there are statistically significant correlations between the total number of events in the two data sets for almost all of the dyads; while Table 2 gives the percentage of correlations significant at the 0.05 level, most are actually

[6]For more detail, see the correlations by individual dyad and the number of events in KEDS and WEIS for each dyad reported in Table 2 in Schrodt and Gerner ([30]:16).

Table 2: Percentage of Dyads Showing Significant Correlations (0.05 level) between KEDS and WEIS Measures

Variable	All Dyads	Dyads with WEIS > 120 Events
Number of events	93%	95%
Net Cooperation	52%	58%
Cooperative events	95%	100%
Conflictual events	55%	84%

significant at the 0.001 level ([30]:16). This is also true for the number of cooperative events reported by WEIS and KEDS. The dyads where significant correlation is not found are usually those where WEIS reports only a small number of events. Because WEIS is based on *The New York Times* and *Los Angeles Times*, while KEDS is based on Reuters, KEDS has approximately three times as many reported events as WEIS and consequently often shows monthly variations in behavior where WEIS reports only zeros. There are substantially fewer significant correlations for the net cooperation value and for the number of conflictual events—roughly 55% of these are significant at the 0.05 level—although if one looks only at the cases where WEIS records an average of at least one event per month (i.e., a total of at least 120 events for the period), the percentage of significant correlations increases to 84% for the conflictual events.

While the correlation between the net cooperation values appears low, the correlation between WEIS and KEDS is substantially higher than the correlation [38] found in a comparison of WEIS and COPDAB. Vincent correlated the total annual dyadic weighted conflict scores reported in WEIS and COPDAB for 128 countries between 1966 and 1978. He found correlations ranging from a high of 0.92 in 1969 to a low of 0.14 in 1972. Vincent's unit of analysis was the country and the annual dyadic behavior for each actor was summed across all of its dyadic partners; Vincent presumably used actor-year totals since a test using all 16,256 dyads would be dominated by cases where no events occurred at all.

Table 3 summarizes Vincent's findings and also shows the results of a similar comparison between the KEDS and WEIS data sets for the Middle East using both the actor-year ($N = 7$) and the dyad-year ($N = 42$). The average correlation between KEDS and WEIS is substantially higher than that reported between COPDAB and WEIS; the variance between years is also less. In the WEIS-KEDS comparison, only 1983 shows an anomalously low correlation; this is in all likelihood due to events in Lebanon and may involve variations in how actors were identified as well as differences in coding the events.[7] Vincent's average COPDAB-WEIS actor-year correlation is 0.66; the average KEDS-WEIS dyad-year correlation is 0.80, and the average actor-year correlation is 0.92, although the latter figure is undoubtedly somewhat inflated by the small sample size. The COPDAB-WEIS average is affected by the unusually low scores in 1972 and 1975, but even if these years are eliminated, the average is only 0.74, somewhat less than the KEDS-WEIS comparison.

We recognize that these tests may not be directly comparable. KEDS and WEIS differ only in their sources, while WEIS and COPDAB use quite different coding schemes and additional variance was introduced in converting them into a single system. Furthermore, COPDAB, unlike WEIS, reportedly depends heavily on regional news sources. The comparison does, however, indicate that the variation between machine and human WEIS-coded event data sets is substantially less than the differences researchers already face in dealing with WEIS and COPDAB.

[7]During 1983 a number of sub-state actors in Lebanon contested the presence of Israeli, U.S., Syrian, and other foreign troops. KEDS codes some of the more persistent of these groups—for example, the Islamic political groups Hizbollah and Amal—as distinct actors; WEIS does not.

Table 3: Comparisons of Event Data Sets by Actor-Year

KEDS-WEIS			COPDAB-WEIS	
Year	By Dyad	By Actor	Year	By Actor
1982	0.74	0.95	1966	0.80
1983	0.59	0.77	1967	0.83
1984	0.76	0.90	1968	0.82
1985	0.84	0.90	1969	0.92
1986	0.80	0.92	1970	0.51
1987	0.91	0.98	1971	0.47
1988	0.96	0.99	1972	0.14
1989	0.93	0.95	1973	0.55
1990	0.69	0.90	1974	0.68
1991	0.80	0.94	1975	0.36
			1976	0.76
			1977	0.87
			1978	0.93
Average	0.80	0.92		0.66
N	42	7		128

Fig. 1 and Fig. 2 compare the time-series for the net cooperation reported by the KEDS and WEIS data sets for the Israel ⇒ Lebanon and Israel ⇒ Palestinian dyads. The obvious general difference between the two data sets is the greater variance in the KEDS series, a consequence of KEDS having more events than WEIS. In a number of months, particularly those after 1989, the KEDS series show activity where WEIS does not.

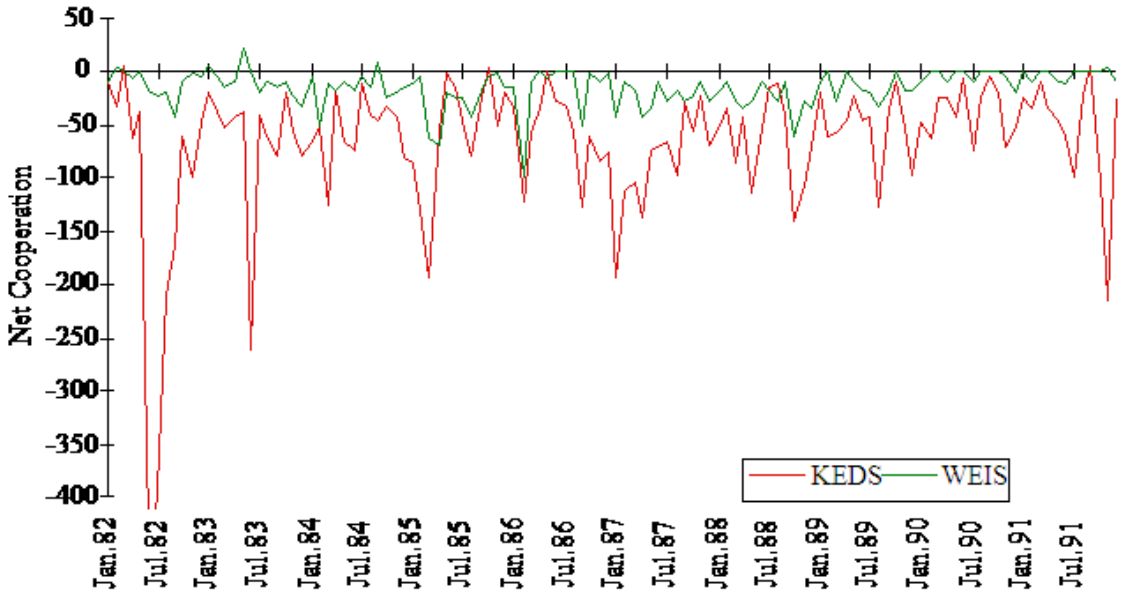

Figure 1: Israel-Lebanese Net Cooperation, 1982-1991.

In Fig. 1, Israel ⇒ Lebanon, both data sets report a significant amount of activity throughout the

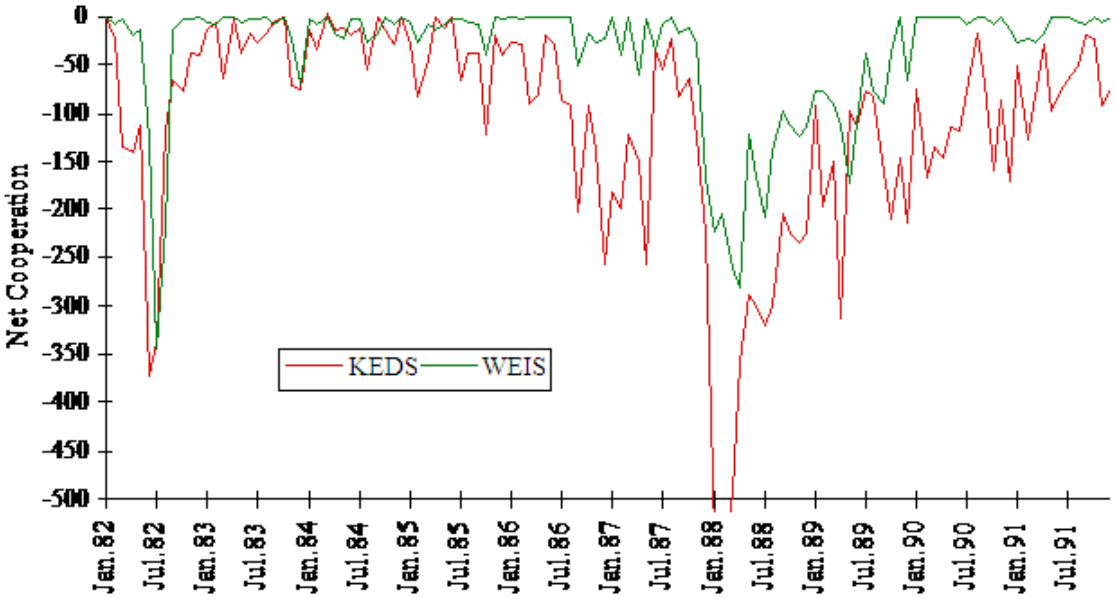

Figure 2: Israel-Palestine Net Cooperation, 1982-1991.

entire period. This series also illustrates a major difference in coding rules. The KEDS series shows a substantial increase in conflictual activity coinciding with Israel's invasion of Lebanon in the summer of 1982 while WEIS shows almost no activity during the same period. The reason for this difference is that WEIS codes the use of force during the invasion—for example, the bombing of West Beirut—as being directed against "ARAB COM" (Arab community, which we recoded as "Palestinian"), and reserves the "Lebanon" designation for actions such as protests and consultations specifically involving the Lebanese government. KEDS, in contrast, codes "Beirut" as "Lebanon" and only codes Palestinians as a target when they or the PLO are explicitly mentioned. This difference goes beyond the question of human versus machine coding and relates to more general event coding issues that often do not have unambiguous solutions. Except for the invasion of Lebanon, the two series covary fairly well, despite the differences in the magnitude of activity; when the WEIS data set shows an increase in activity, this usually is associated with an increase in activity in the KEDS data.

We deliberately included Lebanon as one of the test cases for KEDS because we knew that the various political groups in that country would prove challenging to code. They have. The WEIS coding is consistent with Israel's view of the invasion, since Israel's explicit target was the PLO and Palestinians in Lebanon generally. However, the large number of non-Palestinian Lebanese living in West Beirut in 1982 perceived that they, rather than the Palestinians alone, were under attack by Israel, and some of those Lebanese, notably the Shi'a Moslems, dramatically altered their behavior toward Israel as a result. The 1982 war in Lebanon provides a good illustration of a situation for which the ability of a machine-coding system to inexpensively produce two different versions of the time series, one reflecting each interpretation of the invasion, could be used to an advantage.

Finally, Fig. 2 shows the Israel ⇒ Palestinian interactions. This is the densest of the dyadic series in both the KEDS and WEIS data sets. The two sources track each other quite closely in major events such as the invasion of Lebanon and the onset and evolution of the *intifada*. However,

the WEIS series shows the *intifada* essentially ending in January 1990, whereas KEDS reflects a more general decline that is consistent with the narrative record as well as with statistics on the number of shooting deaths shown in Figure 8 in Chapter 1. We have a sense that *The New York Times* coverage of the *intifada* decreased substantially after the autumn of 1989 when *The Times* shifted resources to covering the political changes in Eastern Europe; the WEIS series is consistent with that interpretation.

Overall, the KEDS and WEIS net cooperation time series show similar patterns. In most cases, the exceptions are found in months where KEDS/Reuters records activity while WEIS/ *Times* does not; those months probably account for much of the unexplained variance in the correlations between the two series. A scattergram of the KEDS versus WEIS series for the U.S. ⇒ Israel dyad, a case with low correlation but a relatively high event density in both series, shows considerably more variance in the KEDS values than in the WEIS values ([30]:21). The cluster of points in the scattergram is arrayed around the WEIS=0 line; this in turn leads to the low overall correlation between the two series.

5 The Machine Coding Process

The ideal system for generating machine-coded event data would take a news source as the input and produce an appropriately formatted event data set as the output. While such a coding system is technically feasible with the integration of existing computer programs, in practice coding an event data set requires several discrete steps. Fig. 3 outlines the process involved in going from machine-readable texts to data that can be analyzed with a statistical program. The remainder of this section will discuss the practical considerations at each step in that process.

Fig. 2.3: Generating Machine-Coded Data

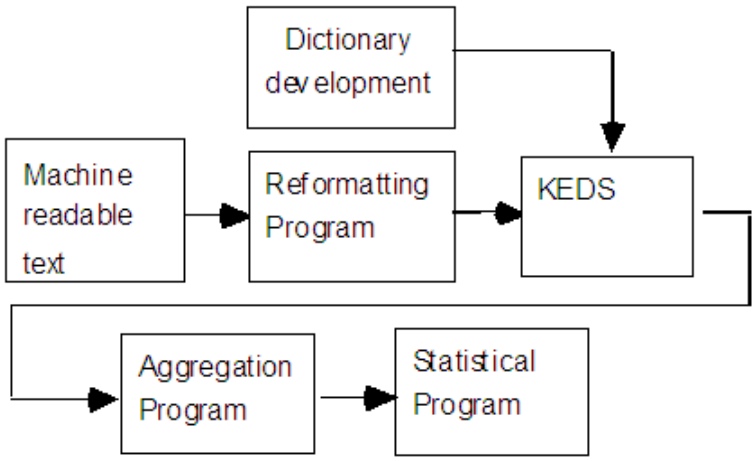

Figure 3:

5.1 *STEP 1: Locate and Reformat a Set of Machine-Readable Texts*

The first step in machine coding is finding a source of machine-readable text. Most of our work during the 1990s used Reuters reports from the NEXIS data service. Originally the text looks like

Fig. 4.[8]
It needs to be converted to the KEDS input format, as in Fig. 5.

```
]]]
]                          LEVEL 1 - 92 OF 991 STORIES
]]                       Copyright 1995 Reuters, Limited
]                          Reuters North American Wire
]]                      <February> 4, 1995, Saturday, BC cycle
]]HEADLINE: <Somali> gunmen demand ransom for U.N. workers
]] BODY:
]   <Somali> gunmen demanded $ 420,000 Saturday for the release of 15 U.N.
]aid workers detained in their house in Mogadishu and isolated from U.N.
]peacekeepers and civilians preparing to evacuate the country.
]]   Gunmen surrounding the house in the shattered capital said the money
]represented salaries owed to 1,010 <Somalis> who had worked for the World
] Food Program at Mogadishu port.
]]   The dispute dates back to the chaotic months before U.S.-led forces, sent to
]sent to end famine, first stormed the beaches of <Somalia> in December
]1992 and seized control of key installations in the city.
]>>>
]
```

Figure 4:

```
950204  REUT0048-001
Somali gunmen demanded $ 420,000 Saturday for the release of 15 U.N.
aid workers detained in their house in Mogadishu and isolated from U.N.
peacekeepers and civilians preparing to evacuate the country.
```

Figure 5:

Over the years, we have developed a set of rather elaborate reformatting programs—written in Pascal or C—to remove all of the irrelevant information found in the news report downloads and reformat the text. (The PANDA project used the macro language in Microsoft's *Word* program to perform the same task.) These programs, which evolved over time as NEXIS changed its formats, can filter the text to find only the lead sentence or alternatively produce individual records for each sentence in the story (with or without sentences that appear to be direct quotes). The filter also converts the English date given in Reuters to the YYMMDD numerical data format used by KEDS and assigns a unique identification number to each record. Finally, it checks the formatted reports of each day for multiple sentences that have almost identical letter counts; this eliminates most of the stories rebroadcast by Reuters to correct spelling errors.

5.2 *STEP 2: Develop the Coding Dictionaries*

KEDS uses large dictionaries of proper nouns (*.actors*) and verb phrases (*.verbs*) to code the actors and events it finds in the source text. The dictionaries originally developed by the KEDS and PANDA projects are the result of about four person-years of working through Reuters lead sentences to identify relevant verb phrases and assign these to an appropriate event code.

[8]The format shown here was found in the dial-up version of NEXIS; NEXIS is now using a Web-based system that presumably looks quite different.

Over the past two years we have been working with a "standard" verbs dictionary that is a composite of several different dictionaries that were developed in conjunction with the KEDS project over the past eight years, including our original Levant dictionaries, the PANDA dictionary, Huxtable's West Africa dictionary [15] and the Pevehouse's Balkans dictionaries [12].[9] After merging these dictionaries, we eliminated assorted phrases that remained from times when KEDS contained bugs[10] and ultimately removed most verb phrases containing more than a half-dozen words on the grounds that such phrases would be repeated only rarely. The resulting dictionary contains most of the verb phrases used by Reuters to describe *international* political events, as well as discard codes for a wide variety of athletic events, natural disasters, and fatal mishaps involving various modes of transportation.

Dictionaries for the coding of *internal* political events present additional challenges and it is not clear whether it will be possible to develop a general-purpose *.verbs* dictionary for this task. When developing *.verbs* dictionaries to code internal events in a disparate set of states—Russia, China, Albania, Colombia, Mexico, Syria, Algeria, Pakistan, and Nigeria—we found that the vocabulary referring to domestic events often varied significantly across regions. First, internal events involve a much larger set of verb phrases than international events and many of these phrases are idiosyncratic to specific states. For example, reports on Colombia and Mexico reflect a great deal of large-scale, quasi-political criminal activity involving the trade in illegal drugs; this type of activity is rare in reports involving the Middle East, Africa, or Europe. Africa, in contrast, involves quasi-political criminal activity involving the smuggling of diamonds that is not found elsewhere in the world. Islamic politics is important in the Middle East and parts of Africa; it is irrelevant to Latin America. When we coded Albania for the 1996-97 period, we encountered a series of events involving the collapse of pyramid investment schemes, followed by an almost complete breakdown of political order, followed by an international intervention that very quickly restored order. That sequence was quite distinct from the civil disorder we coded for Lebanon.

Second, despite the generally consistent style found in Reuters reports, there are certain distinct verb phrases that are employed by the reporters and editors in each geographical region. Each set of regional reports has a few idiosyncratic turns of phrase that we have not encountered earlier.[11] Because these phrases are common, they are discovered very quickly when spot-checking the dictionaries and actually simplify coding when events are reported using a small set of routine sentence structures. Nonetheless, the presence of idiosyncratic phrases means that a dictionary developed in one region will miss some important events if it is used, unaltered, in another region. These errors are almost exclusively false negatives—an idiosyncratic phrase used in one region will almost never correspond to a distinctly different behavior in another region.

Our conclusion from these projects is that anyone attempting to code internal events should invest time in customizing the standard dictionaries. If a sequence of very unusual events has occurred—the collapse of the Albanian financial system, for instance—it may be advisable to develop dictionaries specifically to code that period. Routine internal behavior, on the other hand, can probably be coded reliably using standard dictionaries with some spot-checking for distinct phrasing and regionally specific forms of political activity.

A researcher who is not facing a deadline will be tempted to "tweak" the dictionary vocabulary indefinitely. That will eventually do more harm than good: Tweaking should focus on finding *general* patterns that will occur on multiple occasions in the text, not on expanding the list of

[9] These dictionaries were not developed independently so they contain very substantial amounts of overlap.

[10] More precisely, contained more bugs than it now contains.

[11] When developing a dictionary, there are times when one has a sense of looking over the collective shoulder of Reuters: For example an indicator that Reuters is using inexperienced reporters (or over-worked editors) to cover a crisis are reports containing two consecutive apostrophes (' ') instead of a quotation mark (").

phrases to cover every possible contingency. In our experience, the single most common error made by inexperienced coders working on dictionary development is adding elaborate phrases that may occur only once or twice in any set of source texts. When the source texts generate tens of thousands of events, dozens of such phrases still have an almost no effect on accuracy, particularly if the events are subsequently aggregated using scales or cue categories.[12]

Experienced coders, in contrast, focus their tweaking on three tasks. The first is identifying common new phrases or sentence structures that are due to the style of a particular, if anonymous, reporter or editor. The second task is the reverse of the first—identifying phrases inherited from earlier data sets that cause incorrect coding in the current data. Finally, tweaking involves setting the appropriate levels in the KEDS complexity filter to reject sentences that are likely to be incorrect—for example texts containing an excessive number of verbs or actors, or containing ambiguous words such as GEORGIA.

5.3 *STEP 3: Supplement the* .actors *Dictionaries*

While the verb phrases used to describe international political events differ little across time and geographical regions, there is substantial variation in the political actors, particularly if one is coding sub-state and non-state actors. Consequently, any project focusing on a new geographical region needs to supplement the *.actors* file. These files also need to be periodically updated with the names of new political leaders and, in the post-Cold War period, new states. If internal events are being coded, the required modifications can be quite extensive.

While it is possible to detect new actors by going through the source texts manually, that process is quite labor intensive because of the possibility of an opposition leader or group achieving "fifteen minutes of fame" somewhere in the middle of the data set. To deal with this problem, we have partially automated the process of identifying new actors by using a computer program called *Actor_Filter*.[13] This software goes through a set of text records and, based on patterns of consecutive capitalized words, tabulates phrases that may refer to new political actors. The output of this program is a "keyword-in-context" (KWIC) file of the actors that cannot be found in an existing KEDS dictionary, listed in order of frequency.

The text below shows an example of the first records of the the KWIC index produced by *PoliNER* from a file of events dealing with Bahrain; the actual output preserves the five sentences containing the name but only two are shown here.[14] In the KWIC format, the actors are highlighted with <<...>>. The first two sets of records in Fig. **??** show two common actors that were not already in the dictionary—the Islamic Salvation Front (209 occurrences) and President Liamine Zeroual (182 occurrences)—and also highlights an assortment of other proper nouns such as Air Algerie and Ali Belhadj.[15] The third set of records identifies an actor that is apparently very

[12]One of us recently attended an evaluation of a U.S. government project that had used a human-coded event data. The client began the evaluation by indicating clear irritation over the expense and delay involved in the human coding. The client's mood was hardly improved when everyone present noted that the statistical model using this meticulously coded data had aggregated it into five simple and very broad event categories. Within weeks, the project shifted to machine-coded data.

[13]March 2012: While still available as open-source Java code on our web site, *Actor_Filter* has been superseded in our own work by a Python program *PoliNER* with similar capabilities; this was used extensively in the development of the ICEWS global dictionaries and interfaces with a machine-assisted dictionary development program *CodeCatcher*.

[14]The 2000 text used *Actor_Filter*

[15]These records also show some problems with the consecutive-capitalization rule, particularly when dealing with languages other than English. For example, Reuters does not capitalize the Arabic article "al-," so "Hafez al-Assad" is not seen as a single phrase. The French proper noun "L'Authentique" fails the test because of the contraction; "Liberte" because it is only one word. More sophisticated rules could be developed to deal with these cases, but at the expense of a greater number of false positives. One still has to deal with idiosyncrasies such as the transliteration

common in the texts—the *El Watan* newspaper, which Reuters frequently uses as a source—but which was not coded.

```
GCC [ 4919 ]
** Iraqi newspapers , reacting to statements issued at a two-day summit in Bahrain of six oil-rich states in
the >>> GCC <<< ' Gulf Cooperation Council ( GCC ) , said they were inspired by Washington .
DATE =  010101  ID =  00301030-002  FILE =  all.Bahrain.txt
** The Iraqi media ' s strident attack on the final communique came despite its lack of strong anti-Iraqi
language that has appeared in >>> GCC <<< statements since Iraq ' s 1990 invasion of Kuwait .
DATE =  </ID>  ID =  <FILE>  FILE =  010101

Khalifa [ 1849 ]
** Bahrain ' s Foreign Minister , Sheikh Mohammed bin Mubarak al >>> Khalifa <<< , was to later explain
to the press that the pact would provide for joint defence against external attacks and that it would come
into effect on being ratified by all the States .  DATE =  010102  ID =  00303951-005  FILE =  all.Bahrain.txt
** The United Arab Emirates ( UAE ) President Sheikh Zayed Bin Sultan Al-Nahyan met with visiting Bahraini
Emir Sheikh Hamad Bin Isa Al >>> Khalifa <<< here Thursday on the recently-concluded Gulf Cooperation
Council ( GCC ) summit , the official WAM news agency reported .  DATE =  </ID>  ID =  <FILE>  FILE =  010104

Hamad [ 1502 ]
** Sheik Salman bin >>> Hamad <<< Al Khalifa said the charter , to be voted on in a Feb . 14-15 referendum ,
represents only a foundation for democratic rule .  DATE =  010204  ID =  01517501-002  FILE =  all.Bahrain.txt
** The Emir of Bahrain sheih >>> Hamad <<< bin Isa Al Khalifah will pay the first in the history of relations
between the two countries official visit to Russia on March 19-20 , Itar-Tass learnt from diplomatic sources
on Thursday .  DATE =  010222  ID =  00079769-001  FILE =  all.Bahrain.txt
```

Reuters will refer to an actor using a variety of different phrases. For example, Algerian President Liamine Zeroual might be referred to using any of the following formulations:

```
Algerian President Liamine Zeroual
Algerian President Zeroual
President Liamine Zeroual
President Zeroual
Liamine Zeroual
Zeroual
```

Because of this, most major individual politicians such as chief executives require multiple entries. These are almost invariably subsets of the general structure

```
<nation name> <title> <first name> <last name>
```

and one could presumably use this regularity to further automate the dictionary development process. For example, if the system recognized "Hobbit Liberation Front" as a category [HLF] and encountered the unknown proper noun "Frodo Baggins" in the context, "Frodo Baggins, a leader in the Hobbit Liberation Front," it would tentatively assign "Frodo Baggins" to the category [HLF].

The Reuters editorial style appears to specify that whenever a relatively unknown actor is introduced in a story, he or she is identified by nationality. This characteristic makes the coding of international events relatively easy because one knows the nationality of an actor even if the individual actor is not in the dictionary. It is less helpful in the coding of internal events, where the actor's national identity can generally be assumed from the context of the story. For example in the lead:

"El Watan" ("The Nation") rather than "al-Watan." (This is presumably due the transliteration employed on the masthead of the paper itself, and that in turn was probably chosen by some Spanish typesetter stranded in Algiers in 1923...)

```
Moslem guerrillas killed 14 people overnight in Tabainat village in Blida
province, 50 km (30 miles) south of Algiers, Algerian security forces said
on Monday.
```

both the guerrillas and the people killed are Algerian, but this is not stated explicitly.[16] The KEDS program contains some specialized routines (developed for the PANDA project) that identify geographical location, but we have not used these systematically.

We have created a standard *.actors* dictionary that lists all significant states and international organizations in the international system, as well as major-power political leaders such as U.S. presidents, assorted European prime ministers, and heads of U.N. organizations.[17] When a regional data set is developed, this is supplemented with actors that occur frequently in the *Actor_Filter* output. The rule of thumb that we've been using is to include actors that occur in more than 0.01% of the lead sentences of a data set (e.g. in developing our Gulf data set, which contained approximately 80,000 leads, we added any actors that occurred eight or more times). Because there are a large number of low-frequency actors, the proportion of source texts containing uncoded actors is substantially larger than 0.01%, but the addition of any single actor would change the coding no more than one out of every 10,000 leads. This actually over-states the error, because rare proper names usually contain other identifying information—"Afghan Minister of State for Foreign Affairs Najibullah Lafraie" or "leader of the Tajik armed opposition, Sayed Abdullo Noori"—that permit correct identification. Many of the proper nouns that occur in fewer than 0.01% of the leads involve individuals who are briefly newsworthy, such as hostages or other victims of terrorism, leaders of transient political parties, and the like. Furthermore, because the incidence of proper names has a "long-tailed", rank-size law distribution, just a small number of the names with the highest frequency account for almost all of the actual occurrences.

5.4 *STEP 4: Autocode the Entire Data Set*

Once the dictionaries have been suitably refined, the data should be recoded in fully automatic mode—a process we call "autocoding"—to ensure that the coding rules are consistently applied across the entire data set. If only part of the data set is machine-coded, with occasional records manually "corrected," then inconsistencies will be introduced into the time series that might show up as statistical artifacts later in the analysis. Autocoding also insures that the coding can be replicated by later researchers and can be updated.

The speed of autocoding clearly depends on the speed of the computer, but the following statistics will give some indication of the time involved. During most of the project we were using computers in the 100 Mhz range; these coded around 11 events per second, so an 80,000 event data set such as our Levant and Gulf cases required around two hours to recode. (This processing time includes the evaluation of texts that do not produce events; coding full stories therefore is a bit slower than coding leads.) We recently acquired machines in the 300 Mhz range, and speed increased proportionately to around 45 events per second.

The *potential* speed is dramatically greater than this: KEDS has been compiled on a very old Pascal compiler and our PowerPC microprocessors are actually emulating the older Motorola 680x0

[16] In Gerner et al [10] we describe a similar problem we found when trying to code chronologies from the *Journal of Palestine Studies*. *JPS* assumed certain nationalities were known, so that the statement "Israeli police beat Palestinian demonstrators" was always rendered as simply "Police beat demonstrators."

[17] March 2012: Our standard source now is the file `CountryInfo.txt`, which contains about 32,000 entries on country names, cities, regions and other geographical features, and members of government. This file is available at `http://eventdata.psu.edu/software.dir/dictionaries.html`

processors. When we have translated utility programs from Pascal to C and recompiled native PowerPC code, we've seen increases of speed by factors of 10 to 100. This suggests that machine coding is nearing at the point where a researcher could actively experiment with alternative coding decisions by checking how a dictionary change would affect the entire data set—recoding the data in a couple of minutes—rather than a single text.

Update, March 2012: KEDS was, in fact, replaced by the C++ program TABARI in the early 2000s, and TABARI has generally coded at speeds around 2,000 to 5,000 stories per second—this varies with the size of the dictionaries. In 2009, in conjunction with work on ICEWS, 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 sentence 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 problematic records which at the time could cause the program to crash. With cluster computers increasingly common—at Penn State, we have access to machines with hundreds of nodes, as well as documentation [!]—the speed of coding is now effectively unlimited. [29] discusses this in detail, and a "high-volume processing suite" of utility programs for breaking down and recombining files is available at `http://eventdata.psu.edu/software.dir/utilities.html`

5.5 *STEP 5: Aggregate the Data for Statistical Analysis*

KEDS produces standard event data of the form

```
date <tab> source code <tab> target code <tab> event code
```

For example:

```
960101   IRN   IRQ   032
960101   IRQ   IRN   033
960101   ISR   LEB   220
960101   ISR   PAL   094
960102   JOR   ISR   081
960102   ISR   JOR   081
960102   ISR   JOR   032
960102   JOR   ISR   033
960102   ISR   LEB   220
```

For many applications, this nominal-level (categorical) event series must be aggregated before it can be used by standard statistical programs such as Stata and SAS, or graphical displays such as spreadsheets that expect an interval-level (numerical) series. This transformation is usually done by mapping each event code to an interval-level scale such as the Goldstein or Vincent scales, and then aggregating the data by actor-pair and week, month, or year using averages or totals. It is *possible* to do this aggregation by scripting the data transformation facilities of a statistical program. However, this process tends to be very slow and awkward, particularly when dealing with a large number of actor pairs. We have usually employed a customized aggregation program, KEDS_Count, to automate this process.

So, how much time does this whole process *really* take? We have corresponded with a number of researchers (typically, graduate students) who have successfully used our programs and can provide some general guidelines.

Consistently, the single most difficult and frustrating step in the process is reformatting the raw text into the sentence-by-sentence format employed by KEDS. (Depending on what data services you have available, acquiring a suitable set of texts in machine-readable form can also be problematic.) Sometimes it is possible to employ one of our existing reformatting programs, but even minor changes in the format of the source text will require at least some changes in those programs (and even a single source such as NEXIS will unexpectedly make changes in its format).

At present, none of the major data services—NEXIS, Reuters Business Briefing, and Dow Jones Interactive—provide flexible options for formatting their stories; they assume that their text will be read by humans rather than machines. Fortunately, the formatting is *almost* entirely consistent, but due to editorial and transmission errors, even this cannot be taken for granted. Getting the source text reformatted is likely to take longer than anticipated, particularly if the set of texts being coded extends across a long period of time, and developing a suitable reformatting program is likely to require the assistance of a computer programmer.[18]

Once the texts have been reformatted, the time required for the remaining tasks depends largely on what level of coding error the research project can tolerate. With the *Actor_Filter* program, modifying the *.actors* file can be done in a few hours. The required modifications of the *.verbs* file depend on how well the existing WEIS dictionaries capture the behavior being analyzed, and whether the region being studied has idiosyncratic behavior. But when coding traditional political cooperation and conflict, even the unaltered *.verbs* dictionary will usually correctly identify well over 50% of the events in the source texts. Therefore, once the new political actors have been added, it is possible to get a data set that provides an initial approximation of the final data, and which should show the obvious characteristics of the data. Aggregation of events using KEDS_Count just takes a few minutes, and that program will not need to be modified.

6 A Short Lesson in Linguistics

Before going into some of the details concerning how KEDS converts a sentence to a set of codes, a brief digression is appropriate into some of the characteristics of language—particularly the English language—that make this task challenging. We focus here on the two most important problems: the role of position in determining the meaning of a word in English, and the abundance of homonyms.

6.1 *A Bit of Grammar*

Most native speakers of English remember "grammar" as something they were taught in secondary school. (To U.S. readers: grammatical "parsing" is equivalent to the process of "diagramming sentences" that you probably endured in elementary school.) Grammar consisted of many, many formal rules that were, in the final analysis, quite unnecessary because almost all grammatically correct sentences could be evaluated by applying a much simpler rule—"It sounds right."[19] Many

[18] We would appreciate receiving copies of reformatting programs so that we can post these at the KEDS web site—formats are variable but not infinitely variable, and sometimes it is possible to re-use a program without modification.

[19] The conditions that determine whether a sentence "sounds right" to a native speaker are, in fact, the information that linguists use in determining the grammar of *any* language.

Because the most common grammatical rules are intuitive to anyone who speaks a version of English reasonably similar to standard written English, most people instead associate "grammar" with the arbitrary grammatical rules derived from Latin (an inflected language) that were incorporated into formal English during the 18th century by socially mobile London elites seeking to differentiate their use of the vernacular from that of the masses. The two

non-native speakers of English, in contrast, learn these grammatical rules formally, and as a consequence several projects have found that non-native speakers, because of their more sophisticated understanding of English grammar, are excellent dictionary developers. (The English language model for the VRA parser was developed by Churl Oh, a native speaker of Korean.)

This section will introduce some basic grammatical vocabulary that will be used in the subsequent discussion. We have tried to standardize our use of terms—our reference has been Malless and McQuain [22]—but because some of the problems of machine coding are distinct from those of classical grammar, we may not be entirely consistent with the use of these terms in other contexts. This brief discussion will not exhaust the grammatical issues relevant to machine coding, and a number of additional issues will be presented in later sections, but it will provide a basic introduction.

Consider the following lead sentence:

> `Palestinian President Yasser Arafat accused Israeli Prime Minister Benjamin Netanyahu on Tuesday of intentionally prolonging their peacemaking crisis.`

This has a simple grammatical form that is easily coded by machine. The key components of the sentence are:

> Subject: `Palestinian President Yasser Arafat`
> Verb: `accused`
> Object: `Israeli Prime Minister Benjamin Netanyahu`

The proper noun in the subject—"Arafat" (or the adjective "Palestinian")—is assigned the actor code PAL and becomes the **source** of the event. The proper noun in the direct object—"Netanyahu" (or the adjective "Israeli")—is assigned the actor code ISR and becomes the **target**. The **event** is determined by the verb—"accused"—and is assigned the WEIS code 121. In this instance, the root of the verb is actually identical to the WEIS cue category 120, "Accuse."

In almost all sentences, the subject of the sentence will be the source of the event. However, frequently the event code is determined by the verb and the direct object, with the target of the event being the indirect object. The following lead sentence provides an example of this:

> `The World Bank on Thursday approved a $90 million loan to help Egypt improve the health of its citizens, improve the efficiency and quality of health care services, and ensure equal access for all groups.`

> Subject: `World Bank`
> Verb: `approved`
> Direct Object: `$90 million loan`
> Indirect Object: `Egypt`

In this sentence, both the verb and the direct object in the combination "approved ... loan" determine the WEIS code—071, "extend economic aid."[20] If the direct object was different—"approved

notorious examples of this are the prohibition against split infinitives and the prohibition against ending a sentence with a preposition. Both constructions are completely consistent with the natural underlying grammar of English, but not grammar of formal Latin. Another example is the "that/which" distinction that (which?) is so utterly alien to contemporary English that, in our experience, only the most experienced editors can apply it consistently. The grammatical rules of English required for effective machine coding, in contrast, are so fundamental that many coders who are native speakers of English are surprised to find that the language is in fact governed by such rules.

[20] This is also a good example of the ambiguities of WEIS: an argument could be made that this sentence should be coded as WEIS 052 ("promise material support"), 064 ("grant privilege"), or 082 ("agree to future act"). But despite the fact that the verb is "approved," the event code would not fall into the WEIS 040 "Approve" cue category.

the deployment of troops"—then the code would change, at least in the WEIS scheme. In describing KEDS coding, we refer to the combination of a verb and direct object as the "verb phrase." When the direct object is used to determine the event code, the target code is almost always found in the indirect object.

Both of these examples also contain a number of "subordinate clauses"—fragments of the sentence that contain verbs and nouns but are not the main subject, verb, and object of the sentence. "intentionally prolonging their peacemaking crisis," "improve the health of its citizens," "improve the efficiency and quality of health care services," and "ensure equal access for all groups" are all subordinate clauses.

As in these examples, subordinate clauses usually have the form of an implied subject, an explicit verb and an explicit object. Subordinate clauses are "dependent" on the remainder of the sentence for meaning, whereas the main subject, verb and object form an "independent clause" that is meaningful by itself. In other words, the subordinate clauses can be removed from a sentence and the sentence will still make sense, whereas if the independent clause is removed, the sentence makes no sense.

Event coding focuses almost exclusively on the independent clause, although "issue" coding may use information in the subordinate clause to code the context of an event.[21] Subordinate clauses complicate coding, because they can have the same SVO structure as the independent clause. If the machine coding skips over the subject, verb, or object of the independent clause because of an incomplete dictionary, the system may incorrectly pick up information from a subordinate clause. Subordinate clauses at the beginning of a sentence, such as

> `Despite the low probability of an Iraqi attack on Israel, the State Department on Saturday advised Americans to avoid travel to the West Bank and Gaza and to use caution in Jerusalem.`

are also frequently miscoded.[22]

Subordinate clauses need to be distinguished from *compound sentences*, which contain two or more independent clauses connected by conjunctions, and therefore generate multiple events. For example, the sentence

> `Lebanese leftists demonstrating outside parliament on Tuesday burned a U.S. flag and shouted slogans supporting Iraq in its confrontation with the United States.`

contains at least two events: Lebanese burned a U.S. flag (WEIS 133, "symbolic protest") and Lebanese supported Iraq (WEIS 042, "endorse policy").

The KEDS system generally does quite well with the compound sentences found in Reuters leads. If an actor is not found between the conjunction ("and") and the verb following the conjunction, KEDS inserts the first actor between the conjunction and verb, and then the second independent clause can be coded using the same rules applied to a simple sentence. When we have compared KEDS coding to that of humans, KEDS is more likely than a human coder to find all of the events in a compound sentence, which partly compensates for the errors it makes because of subordinate clauses.

[21] For example, the subordinate clause (and indirect object) in the sentence "Syria has backed Iraq's calls to change the makeup of a U.N. weapons inspection team, saying the demand did not clash with U.N. Security Council resolutions." places Syria's support for Iraq in the context of U.N. deliberations.

[22] The fact that the phrase "State Department" is used rather than "United States Department of State" is also not helpful.

6.2 *The Problem of Disambiguation*

In formal languages such as those used in computer programming, "words" (sets of characters) are associated with a single meaning or a small set of related meanings. While this is often true when dealing with natural language—for example Reuters leads containing the English words *accuse* and *deny* are almost never coded incorrectly—there are exceptions.

In fact, the English language contains *lots* of exceptions. As Pinker [27] notes, English is a language containing an extraordinary number of homonyms—words that sound identical but have different roles and meanings depending on how they are used in a sentence, and the context of the sentence. The problem of determining the intended meaning of a word, and the correct part-of-speech—noun, verb, adjective, and so forth—is called "disambiguation," and is one of the major features of automated natural language processing systems.

English allows words to have radically different meanings depending on their context. For example, the noun "bat" can refer either to a wooden (or aluminum) cylinder used in the game of baseball, or to a small flying mammal. It can refer to the act of batting ("at bat"), or to the unrelated action of blinking ("bat an eye"). Idiomatic uses are even more diverse: "go to bat for" means defending or interceding; "right off the bat" means immediately; "bats in the belfry" is an insult on the target's reasoning ability. The mixing of foreign words introduces additional possibilities: "bat mitzvah" is a Hebrew phrase referring to a girl's coming-of-age ceremony. Any of these uses might be encountered in an English-language text, and multiple uses might be found in a single sentence ("Sarah lowered her bat when she saw the bat flying toward the pitcher.").

Disambiguation of word meaning is less of a problem in coding news stories because most of the time one is dealing with a relatively small set of contexts. This contrasts to the coding of open-ended questions ("What is the most important problem in your life?") where the context is less predictable. Nonetheless, as we will discuss below, disambiguation can be a problem even in texts dealing with political events, most notably when words such as "attack" or "struck back" are used to refer either to rhetorical or military activity.

From the perspective of parsing, disambiguation is likely to involve the assignment of parts-of-speech. This is due in part to the fact that English is primarily an "isolating" language where the grammatical role of a word can change depending on its position in a sentence. In contrast, in "inflected" languages—for example Latin, Russian, or American Sign Language—the root word is usually modified by prefixes, suffixes or vowel changes to indicate that it is being used for a different purpose. Vocabulary in English is further complicated by the fact that English is derived from an inflected Germanic language but evolved over a millennium into a language that is now largely isolating.[23]

For example, in the phrase

```
The treaty was broken
```

the word "broken" is a verb, while in the sentence

```
The diplomats discussed the broken treaty,
```

the word "broken" is an adjective. The distinction is made solely on the basis of the relation between "broken" and the other words in the sentence.[24]

In Latin, either of the following phrases correspond to the English "Man bites dog":

[23] Vestiges of inflection remain at quirky points in the English language—for example "grammatically correct" English retains (barely...) the distinction between the nominative *who* and the accusative *whom* while dropping that distinction in *you*/*ye*, and the inflected "-*ed*" is used to indicate past tense.

[24] Disambiguation in KEDS has been complicated by an early—and bad—feature built into the sparse parser: KEDS ignores the articles "the," "an," and "a." KEDS was initially designed according to principles used in automated

```
Homo canem mordet
Canem homo mordet
```

The order of the subject and object are irrelevant; to create the sentence "Dog bites man," the nouns themselves must be changed:

```
Canis hominem mordet
Hominem canis mordet
```

English, in contrast, uses the same word whether a noun is a subject or object[25] and the role of the noun as subject versus object is determined by its position. Those positions can be changed by the use of passive voice:

```
The dog was bitten by the man
```

but the words "dog" and "man" are still unchanged.

Because of the lack of inflection, the sparse parsing approach used in KEDS will sometimes incorrectly identify the direct or indirect object of a verb:

European Community governments agreed in principle on Monday to a *German* proposal for *EC* financial aid to *Israel* to help it through the Gulf War.
EEC GER 071 (Extend economic aid)

This usually arises when there is an actor in the *direct* object of a verb but the correct target of the event is in the *indirect* object. If the above sentence was phrased "EC governments agreed to give financial aid to Israel" or "EC governments agreed to give Israel financial aid," KEDS would have coded it correctly.

Words can also change from verbs to nouns without modification: Consider

```
I plan to drive to the store, then wash the car
```

and

```
When Jill returned from the car wash, she parked her car in the drive.
```

The only indication in the second sentence that "wash" and "drive" are nouns rather than verbs (as in the first sentence) comes from their position in the two prepositional phrases. In the memorable terminology of Bill Watterson's *Calvin and Hobbes*, "Verbing weirds language."

In our work with Reuters reports on the Middle East, two words stand out as particularly problematic: "force" and "attack." Both words can be used either as nouns ("A guerrilla force launched an attack") or as verbs ("Rebel radio said guerrillas would attack in order to force concessions") and occur frequently in reports about military conflict. Not only is the part of speech ambiguous, but so is the meaning: "force" and "attack" can be used to refer both to verbal actions (persuasion and criticism) and to instances of physical violence; both uses are common in Reuters. In our

indexing and classification, where articles top the list of "stop words" that are ignored because they contribute nothing to the content of a sentence. (Remove the articles from an English sentence and its meaning is rarely affected; many languages either use no articles at all, or use far fewer than English.) Unfortunately, as KEDS evolved into a system based on parsing, it was unable to use the very valuable syntactic information provided by articles for the disambiguation of nouns/adjectives and verbs. "the broken" cannot be a verb, nor can "an attack" or "a force." **March 2012:** This problem was corrected in TABARI , which also considered mid-sentence capitalization to avoid identifying names such as the common Arab name "Said"—as in the noted humanist "Edward Said"—as verbs, but noun/verb disambiguation remains a major issue.

[25]English retains this distinction in pronouns: "I" versus "me," "she" versus "her," and so forth.

dictionaries, a large number of patterns are associated with each of these words to try to distinguish the noun usage from the verb usage. "arms," "battle," "fire," "help," "order," "plan," "pledge," "strike," and "support" are other examples of words that are used both as verbs and as nouns.

Another problematic situation arises from very short, common words such as "by," "in," and "to." These can serve a wide variety of different roles depending on context; they are sometimes called "marker words" because they simply indicate some transition in the sentence, but not a specific transition.[26] For example, "by" is a useful marker for passive voice, but the *Random House College Dictionary* [13] also lists 28 additional meanings for "by." There are 31 distinct meanings for "in" and 25 meanings for "to." The words "in" and "to" cannot be treated solely as if they were prepositions. A fluent speaker disambiguates these uses by context without even thinking—for example, "The negotiators are going to the meeting to be held in the village in a week"—but a computer does not have this capability.

What does this mean for KEDS and machine coding generally? KEDS's dictionaries are primarily oriented toward looking at the location of various words with respect to each other because this is very important for disambiguating words in English. Dictionary development involves a lot of effort in finding phrases that must be eliminated to avoid confusion with alternative meanings of the word—"null-coded" in KEDS terminology—or otherwise assigned an interpretation distinct from the root. This is not a feature of machine coding but rather a feature of the English language. While more sophisticated parsers would be able to deal with some of these problems, the bottom line is that some sentences in English cannot be unambiguously interpreted based solely on the words they contain. On the positive side, working with a machine coding system gives one a much deeper appreciation of how the grammar of English actually functions in the real world, as distinct from the much more simplified grammar we were taught in grade school.

7 How KEDS Evaluates a Sentence

The input to KEDS is a file containing a set of sentences, each prefixed with a date[27] and other identifying information, and followed by a blank line:

```
 980216 REUT-0001-01
Egypt's President Hosni Mubarak warned in an interview published on Monday
that the situation in the Arab world could deteriorate if the United States
attacks Iraq for failing to comply with weapons inspections.

980216 REUT-0002-01
Iraqi Foreign Minister Mohammed Saeed al-Sahaf said on Monday that he was
going to Paris only to take a message from President Saddam Hussein to French
President Jacques Chirac about Baghdad's showdown with the United States.

980216 REUT-0003-01
Israeli businessmen, Jordanian officials and foreign bankers agreed on Monday
that the Israeli-Jordanian peace treaty was not producing economic dividends
quickly enough.
```

[26] "de" plays a similar role in French; "fi" in Arabic.

[27] KEDS uses a two-digit year but "windows" the digits 00 to 10 to 2000-2010. A replacement program with 4-digit years should be available well before 2010.

To code a sentence, KEDS goes through the following operations:[28]

7.1 *1. Word Classification*

The source text is first converted to a standard form. All letters are changed to capitals[29] and commas are delimited with spaces. KEDS then checks each word in the text to see if it occurs in the actor, verb, and classes (synonyms) dictionaries. If the word is found, it is assigned the appropriate type (e.g. actor, verb, pronoun, conjunction); otherwise it is designated as untyped. Most of the subsequent parsing operations deal only with the words that have been classified by type.

7.2 *2. Process Local Grammatical Structures*

This processing includes assigning actor identities to common nouns (PANDA's "agents"), assigning the references to pronouns, using titles to reduce two actor references to a single actor (e.g. "Israeli Prime Minister Rabin" is reduced to a single reference to Israel), recognizing compound noun phrases and eliminating subordinate phrases delimited by commas. If customized "rules" are being used—for example, a general rule to deal with the English-language passive-voice construction—they are also applied at this point. The "complexity filter" conditions are applied at this point, and sentences that appear too complex for KEDS to code are written to a separate file rather than coded.

7.3 *3. Event Coding*

The program next attempts to match the patterns associated with each verb in the sentence to phrases in the *.verbs* dictionary; each of those phrases, in turn, is associated with an event code. Patterns typically distinguish between direct objects, as in the distinction between "promised military aid" and "promised to veto." If a verb phrase corresponding to an event is identified, the program finds the source actor and target actor associated with the verb. The source is usually the first actor in the sentence; the target is usually the first actor following the verb, provided that actor has a code distinct from the code of the source. If no such actor is found, the program then looks for an actor prior to the verb that has a code distinct from the code of the source. If the source or target is a compound phrase, it is expanded into multiple events. Only the first verb corresponding to an event is coded unless the sentence is compound (i.e. contains a conjunction not associated with a compound actor), in which case each clause of the compound sentence is checked for an event.

7.4 *4. Display the Information*

Following the processing, the main display of KEDS will show the source text along with its date and identification number, the coded events and some summary statistics. The main display can also show the parts of speech assigned to various words: actors are shown in red, verbs in blue, agents in green, pronouns are replaced with their references and text eliminated by subordinate

[28] This description is a summary and does not discuss a number of idiosyncratic exceptions and options employed by the parser; further detail can be found in the KEDS manual.

[29] This was another less-than-optimal design decision, albeit one based on the formatting of older texts in NEXIS, some of which were ALL CAPS and the dictionaries had to accommodate this contingency. In addition, ignoring capitalization slightly simplified processing, but complicates disambiguation and the recognition of unknown words: A capitalized word in the middle of an English sentence can only be a proper noun or a formal title.
March 2012: This problem was corrected in TABARI.

phrases or null codes is shown crossed out. This display is particularly useful when one is trying to figure out why KEDS has made a set of coding decisions and in determining unintended word matches.

A second window lists all of the events found in the source text. This display can be customized, but it typically includes the source and target, their agents (if these are being coded), the event code, the English interpretation of the event, and the specific text matched to generate the code. Incomplete events—those missing a source, target or event—are displayed in gray; complete events are in black.

If KEDS is being used for machined-assisted coding, the individual events can be edited; otherwise the events are simply written to an output file. The coded output can be formatted in a variety of ways, including tab-delimited formats for use in database, spreadsheet, or statistics programs.

8 Dictionaries

KEDS was designed as a general-purpose coding system rather than a WEIS-coding program, so most of its coding decisions depend on its dictionaries. All of KEDS's files are stored externally as simple ASCII ("text") files so that they can be edited using a word processor and transmitted using electronic mail. The verbs and actors, as well as their associated codes, also can be added, deleted or changed using a dialog invoked from the main program menu as coding is being done. The editing routine also keeps track of the coder who added each phrase to the dictionary and the date the phrase was added.

A standard input format is used for the dictionaries. Words are entered in upper case; codes for actors and events are enclosed in square brackets []. If two words must be consecutive, they are connected by an underscore; if the two words are separated by a space, other words can intervene. For example, the text "agreed to provide a loan" can be matched by the pattern AGREE LOAN but not the pattern AGREED_TO_LOAN.

If a word in a KEDS dictionary ends in a space, it is used as a stem. Stemming refers to the process of reducing different forms of a word to a single root by removing suffixes[30]:

```
 ACCEPT ⇐ACCEPTS ACCEPTED ACCEPTING
SYRIA ⇐SYRIA'S SYRIAN SYRIANS
```

KEDS handles stemming by matching patterns from the beginning of the word; a word is considered to match if every character in the root matches. In other words, "SYRI" will match all four forms of "Syria" but it will not match "Syracuse." Long phrases are searched before shorter ones: for example "SIGNALLED," "SIGNED," and "SIGN" are checked in that order. An underscore character after the word means that the word will match only if it is followed by a space, so the root "OF_" will only match the single word "of" whereas "OF" would match "of," "offer," and "official."

Stemming had two advantages in the early stages of our work when the dictionaries were relatively small. First, most regular verbs had to be entered only once. The exception to this occurred when a verb root could be mistaken for a noun—for example "FIRE"—in which case multiple forms

[30]The more general task of finding roots is called "lemmaization." This goes beyond stemming to employ algorithms dealing with how a language converts the root to other forms—for example, the plural of a noun or the third-person-plural form of a verb—whereas KEDS's "stemming" only applies a very simple rule to achieve this. Thus a lemmaization system would not only associated "speak" and "speaks"—which can be handled by stemming—but also the irregular forms "spoke" and "spoken." Stemming is generally effective in English, where verbs are primarily modified with suffixes. Lemmaization, in contrast, as almost essential in languages with more complex transformation rules, for example Arabic, German, and Hebrew.

of the verb had to be entered explicitly. Second, nouns related to a verb—for example "MEETING/MEET" or "ACCEPTANCE/ACCEPT"—would trigger a correct classification even when an uncommon verb in the source text was not in the dictionary.

With more extensive dictionaries, however, stemming is the most frequent cause of wildly inaccurate coding errors. For example, when we were reports from the Middle East for November 1993, the usually problem-free verb "BEAT" matched "Beaty," the name of an American businessman released from prison by Iraq. If we were writing KEDS today, we would probably incorporate a facility for explicitly defining regular verb constructions and noun endings (e.g. plurals and adjectival forms) and drop stemming.

The KEDS dictionaries we have used to code international events in the Middle East contain about 650 actors and 3500 verb phrases. The PANDA project used slightly larger dictionaries: 880 actors, 4300 verb phrases, and 200 agents. Earlier work [9] on automated processing of English-language reports of political violence indicated that dictionaries on the order of 5000 phrases are necessary for relatively complete discrimination between political events, so these KEDS-compatible dictionaries are probably close to having a relatively complete vocabulary. Despite the fact that PANDA is coding the entire world while the Kansas project codes only the Middle East, the actor dictionaries are about the same size because a fairly complete list of actors is required to identify potential targets of events. Dictionaries for coding internal events are somewhat larger, although the number of additional phrases required is usually in the tens or hundreds, not the thousands.

8.1 *Proper Nouns: Actors*

An *.actors* file contains proper nouns and associates each of these with a code:

```
 ABU_SHARIF [PLO]
ACQUINO [PHL]
AL-WAZIR [PAL]
AMMAN [JOR]
AMNESTY_INTERNATIONAL [NGO]
ANKARA [TUR]
ANTIGUA [ATI]
```

Multiple nouns can map to the same code; for example "ISRAEL," "ISRAELI," "ARENS," "PERES," "RABIN," "TEL_AVIV," and "SHAMIR" all have the code ISR.

In some circumstances, it is useful to have a single phrase generate multiple actor codes. In KEDS this is indicated by separating the actor codes with a slash:

```
 EAST_AND_WEST_GERMANY [GME/GMW]
NORTH_AND_SOUTH_KOREA [KON/KOS]
G7 [USA/GMW/FRN/ITL/UK/JAP/CAN]
```

Actors also change over time. The two most notable instances of this in our Middle East data are Boutros Boutros Ghali—who appears both as Egypt's foreign minister and as Secretary General of the United Nations—and the changes surrounding the collapse of the Soviet Union. These situations are dealt with by indicating different codes for different time periods:

```
MOSCOW [USR (<901225) RUS (>901226)].
```

8.2 *Common Nouns: Agents*

The early event data sets such as WEIS and COPDAB were state-centered and made little or no distinction among substate actors. This convention is also found in many Reuters leads, where the names of the states are used to refer to actions of governments or foreign ministries: "Israel accused Syria ...." However, in many circumstances it is useful to differentiate the agent responsible for an event, for example distinguishing "Israeli soldiers," "Israeli police," and "Israeli settlers." This is particularly important in the PANDA coding scheme, which deals with many internal political activities such as strikes, elections, and protests.

The KEDS program can deal with this in either of two ways. If one is coding a small number of countries, the agents can be coded explicitly in the dictionaries:

```
 ISRAELI_POLICE [ISRPOL]
ISRAELI_SETTLERS [ISRSET]
ISRAELI_SOLDIERS [ISRMIL]
```

While this approach leads to longer dictionaries, it also allows the secondary codes to be very specific: for example major opposition parties can be assigned distinct codes while minor parties are lumped together in a general OPP category.

For the PANDA project, however, a more sophisticated approach was used that employed a single list of substate actors called "agents." In some cases, an agent is implicit in a proper noun—for example "GEORGE BUSH" was president of the United States—and these codes are specified in the actors dictionary. In many other cases, the agent is identified by a common noun:

```
 AGENT: DISSIDENT [OPP]
AGENT: ELECTORATE [CON]
AGENT: EMIGRANT [REF]
AGENT: EMIGRES [REF]
AGENT: ENVIRONMENTALIST [ENV]
```

KEDS attempts to assign an actor identity to all agents, using the following priority:

1. Implicit agents: GEORGE BUSH [USA:GOV]

2. <actor><agent>: FRENCH POLICE

3. <agent><preposition><actor>: POLICE IN DAMASCUS

4. an actor found within ±2 words of the agent

If none of these patterns occurs, the agent is assumed to have no explicit actor and is then treated as an actor when identifying sources and targets. The statement "Police fought demonstrators" will generate an event of the form

```
*** POL *** DEM
```

where *** is KEDS's code for an unknown actor. Alternatively, a default value could be assigned to an unknown actor: For example during the Palestinian *intifada*, "Police fought demonstrators" in the West Bank almost always meant Israeli police fighting Palestinian demonstrators. More recently, however, this statement could—and with increasing frequency, does—also refer to Israeli police fighting Israeli demonstrators, or Palestinian police fighting Palestinian demonstrators. (We will know that a resolution of this conflict has truly arrived when we see Palestinian police fighting Israeli demonstrators.)

8.3 *Verbs*

The *.verbs* file contains verb phrases and their associated event codes. This includes both simple verbs (e.g. VISITED) and verbs plus direct objects (e.g. PROMISED FUNDS).

The following is an example of an entry in the *.verbs* dictionary:

```
 ACCEPT
- * PROPOSAL [081]
- PROPOSAL WAS * [081]
- * CHARGES [013]
- * FORMULATION [042]
- * INVITATION [082]
```

In this example, the root verb is ACCEPT; with stemming this will match ACCEPT, ACCEPTS, and ACCEPTED. The phrases that start with "-" are the patterns associated with ACCEPT and their codes; the "*" indicates where the verb itself should appear. In the example, ACCEPTED PROPOSAL will be coded 081 ("agree" in the WEIS coding system) while ACCEPTS FORMULATION will be coded 042 ("approve" in WEIS).

Patterns usually involve direct objects or modifiers such as NOT. The key to this scheme is ensuring that phrases are associated with a transitive verb rather than indicators of tense such as HAS, WILL, IS, WAS, WERE. The important verb in a phrase will often be an infinitive; for example in WILLING TO NEGOTIATE, the verb is NEGOTIATE. Pattern matching stops at any conjunction; this prevents a pattern from matching the direct object of another verb in a compound sentence.

Patterns can also specify where the source and target are found in relation to the verb and associated words; these are indicated by "$" and "+" respectively. For example, the pattern

```
ADVISE - + WAS * BY $
```

would make the correct source and target assignment on the passive construction "Egypt was advised by the United States."[31] The symbol "%" specifies that a *compound actor* should be assigned to both the source and target. This is typically used when dealing with consultations to indicate that the subject of the sentence contains both the source and target.

```
 Israeli and Palestinian officials will meet on Sunday to discuss some of
the issues still blocking a peace agreement, Palestinians said on Saturday.

Israeli businessmen, Jordanian officials and foreign bankers agreed on Monday
that the Israeli-Jordanian peace treaty was not producing economic dividends
quickly enough.
```

[31] Passive voice was originally handled on a case-by-case basis using the $ and + tokens. In later versions of the program, passive voice was handled by a general grammatical rule.

8.4 *Pronouns*

Pronouns occur frequently in Reuters:

```
Turkey believes Iraq and Syria can cope with a decrease in vital water but
they have lodged a protest with Ankara.
```

THEY refers to "Iraq and Syria," but the program must determine this in order to code the second clause of the compound sentence correctly. Determining the correct reference for a pronoun can also bring an actor to a point in the sentence where it will be correctly matched by a pattern.

Ascertaining the references of pronouns is a very general problem in parsing. Unfortunately, a reference often cannot be resolved on a purely syntactic basis. In the sentence "Baker will meet with Mubarak when he goes to Geneva" the pronoun "HE" could refer to either "Baker" or "Mubarak" depending on who is going to Geneva. Sophisticated parsers (and humans) can often use semantic information to resolve references. In the sentence "John took the cake home and ate it," the word "it" refers to "cake" because one does not eat "home," whereas in the sentence "John took the baseball bat and broke it," the word "it" refers to "bat" rather than "baseball." Nonetheless, without very good disambiguation of parts of speech, the structure of the two sentences appears identical.[32]

KEDS does not use semantic information, but in most Reuters leads this is not required. Instead, a simple set of rules are used (Table 8.4).

These rules are least effective on the pronoun IT because that word often refers to an activity

HE SHE IT	assign the first actor in the sentence
ITS	assign the first actor prior to the pronoun
THEY	assign either the first compound actor if one exists or else assign an actor followed by a word ending in 'S' or an agent, for example "Syrian soldiers" or "Israel police."

rather than an actor. For example in the sentence "Police had said the rally was banned, but did not prevent it from taking place," "it" refers to "rally." In the sentence "Interfax news agency, citing unnamed sources, said it would take four days for the troops to deploy," "it" has *no* reference, but instead serves as a placeholder for the implied subject of the sentence.

9 Other Parsing Features

9.1 *Compound Sentences*

When multiple verb patterns are found in a sentence, events are prioritized by:

- left to right order of verbs in the sentence
- length of patterns within a verb's pattern list

Events are coded from only the first verb in a sentence unless the sentence is compound or the first verb has been designated as "subordinate." (In KEDS terminology, a "subordinate" verb is one that is coded only if no other verb can be found to code: see discussion in section 2.7.2.) KEDS recognizes compound sentences generated by the conjunctions AND and BUT; this is determined

[32]The structures are not in fact identical—"cake" is used as a noun whereas "baseball" is used as an adjective, and a pronoun reference must be to a noun. This level of parsing, however, goes far beyond the capabilities of KEDS.

after any AND found in compound a noun phrases such is "Baker and Mubarak" is eliminated. In a compound sentence, the source of the event is not changed unless an actor occurs immediately after the conjunction.

KEDS recognizes compound nouns of the form

```
 <actor1> AND <actor2>
<actor1> , <actor2> , ...<actorn-1> AND <actorn>
```

and does the appropriate duplication of events.[33] For example, "The United States and Egypt approved of efforts by Israel and Jordan" would generate the four events

```
 USA <APPROVED> ISR          USA <APPROVED> JOR
UAR <APPROVED> ISR          UAR <APPROVED> JOR
```

The most common error that results from these rules occurs when a compound noun is in the sentence but the nouns are not actors

```
Teachers demanding better pay and improved benefits marched again through
the streets of the center of Amman on Wednesday.
```

The AND will cause this to be interpreted as a compound sentence. Conversely, one will occasionally find compound sentences where the phrases are separated with the compound noun structure

```
<actor> AND <actor>
```

In this instance, the second phrase in sentence will not be interpreted as being compound.

```
The United States confirmed on Friday that it is prepared to respond positively
to a U.N. appeal for more emergency food aid for North Korea and U.S. officials
said the contribution would be $6 million.
```

The following text illustrates two other problems arising from conjunctions.

```
An artillery battle between Israeli and Palestinian forces in Beirut broke
a 24-hour-old ceasefire today as President Reagan agreed in principle to send
U.S. troops to help evacuate Palestinians from the city.
```

ISR USA 223 (military engagement)
PAL USA 223 (military engagement)

In this example, the phrase "Israeli and Palestinian" was identified as a compound actor corresponding to the verb phrase BROKE CEASEFIRE and the target REAGAN.

The erroneous coding occurs for two reasons. First, the lead reports two events—a battle and a U.S. agreement—separated by the conjunction AS. While AS is a perfectly legitimate English conjunction, it also has 26 other meanings [13], and therefore it is not in KEDS's list of conjunctions. If AND or BUT had been used as the conjunction, or if AS was recognized as a conjunction, KEDS would have correctly picked up "Reagan agreed to help Palestinians." But this did not occur in the source text.

We have developed a rule (see section 2.6.6) for handling these "semi-conjunctions":

[33] In order for a compound to be recognized, the actor must follow immediately after the AND: the phrase YITZHAK SHAMIR AND A TIRED JAMES BAKER would not code as a compound because of the intervening adjective TIRED This type of phrase is relatively infrequent in Reuters.

```
 CLASSES
<SEMICONJ> = AS_ WHILE_ WHEN_

RULES
<SEMICONJ>_<ACTOR> -> <CONJ> <ACTOR>
```

This simply substitutes a generic conjunction for the semi-conjunction; that conjunction in turn will trigger the processing of a compound sentence.

The second problem in the text involves the verb pattern. While a number of verbs have patterns that are specifically looking for compound subjects—most commonly meetings and military engagements—BROKE CEASEFIRE does not have such a pattern, because this activity is typically done by only one actor. In this example, the ideal pattern would have focused on "battle between" and assigned a symmetric event with ISRAEL_ AND_PALESTINIAN as the compound actor. This would produce the correct coding despite the fact that BATTLE is actually a noun and the subject of the sentence, rather than a verb; this is a case where KEDS would do the correct coding for the incorrect reason.

9.2 *Paired and Subordinate Codes*

The WEIS coding scheme frequently generates symmetric events of the form

```
 <Actor1> <Event1> <Actor2>
<Actor2> <Event2> <Actor1>
```

For example, a meeting between Israel and Egypt at a neutral site would generate the pair of events:

```
 ISR 031 UAR     (meet with)
UAR 031 ISR     (meet with)
```

A visit by a Jordanian official to Syria would generate the event pair:

```
 JOR 032 SYR     (visit; go to)
SYR 033 JOR     (receive visit; host)
```

In KEDS these paired events can be coded automatically by using a pair of codes separated by a slash ; for example

```
 FLEW_TO [032/033]
```

would code the visit/receive pair.

KEDS dictionaries can also set priorities when multiple verbs are found in a sentence. A *subordinate* code indicates that a verb is only to be coded if no other events are found. When a phrase with a subordinate code is encountered, KEDS continues to search for other verb patterns in the sentence rather than stopping. This is typically used for verbs of attribution such as SAID or REPORTED. For example, in coding "George Bush said he rejected Syria's assertion . . . " the relevant event is USA <REJECTED> SYRIA rather than USA <SAID> SYRIA. By associating GEORGE_BUSH with the pronoun HE and using a subordinate code will allow this to be coded properly.

9.3 *Deletion of Subordinate Phrases Delimited by Commas*

In Reuters leads, short phrases delimited by commas and phrases between a comma and the end of the sentence are usually irrelevant to the coding:

> President Mubarak, in a grim warning underlining Egypt's deepening economic crisis, will request emergency assistance from the IMF, the official UAE news agency said on Thursday.

These phrases are eliminated by the parser if the number of words between the commas is greater than two and less than or equal to ten; the minimum allows the preservation of comma-delimited lists. The maximum and minimum length for an eliminated phrase can be set by the coder and commas inside numbers such as "10,000" do not trigger this feature.

Reuters will occasionally introduce an event by using a subordinate phrase which indicates why the event is important:

> A surprise setback today hit efforts to end the war between *Israeli* forces and Palestinian guerrillas when Syria rejected the idea of the entire *Palestine Liberation Organisation* moving to its territory.

ISR PAL 111 (Reject; turn down proposal) Unfortunately, this sentence construction forces the event to occur far from the beginning of the sentence and provides ample opportunity for an extraneous actor to be associated with the verb. Such sentences are unlikely to be in SVO format and therefore will probably code incorrectly; they can, however, be readily eliminated using the complexity filter (see section 2.7.6). A story such as this one probably would have been preceded by a story with an explicit SVO lead about the Syrian rejection, so the underlying event would have been coded even if this sentence could not be coded.

9.4 *Null Codes and Stop Words*

The null code "- - -" is used to eliminate phrases that would otherwise be confused with actors or verbs. For example, the phrase "Israeli-occupied West Bank and Gaza" will generate both the ISR and PAL codes as actors. By adding the null code

```
ISRAELI-OCCUPIED [---]
```

only PAL is generated as an actor. The phrase "The head of Lebanon's Catholics" would generate a verb identification for HEAD, a common verb,[34] but including the phrase HEAD_OF_ with a null code eliminates this problem.

Null codes have proven surprisingly important in refining the coding of the system, particularly given the aforementioned propensity of English to use words as both nouns and verbs. Some of these words present insurmountable problems, but null coding can eliminate many troublesome phrases. For example, some of the ambiguity in the words ATTACK and FORCE can be removed by null-coding the combinations

```
 AN_ATTACK [---]
A_FORCE [---]
THE_ATTACK [---]
THE_FORCE [---]
```

[34] For example, "Egyptian President Mubarak headed for a meeting with ..."

In these instances, the presence of an article indicates that the word is being used as a noun rather than a verb. This still does not resolve situations where the word is modified by other adjectives (for instance, "a rebel force"), but it's a start.

9.5 *Issues*

KEDS can code up to nine sets of "issues": these are typically sets of words or phrases identifying the context or domain of an event. For example, the PANDA "Issues" variable begins with the phrases:

```
 ABORTION [T]
AIDS_ [E]
ANCESTRAL LAND [N?]
APARTHEID [H!]
ASYLUM [H]
BALLOT [G?]
BAN_THE_DEATH_PENALTY [P?]
BANKING [F!]
```

Issue phrases can be coded as dominant (!) or subordinate (?), or given a numerical priority between -254 and 255. The code for the issue with the highest priority will be assigned to the event. An issue can also default to the code of another variable; for example, PANDA's "Location" variable defaults to the source if no other location is found.

9.6 *Discard and Complex Events*

Some news reports involve multiple international actors but no political events. Sports events are especially problematic given the propensity of Reuters to use national identities and martial metaphors in describing the athletic contests, particularly soccer ("Algeria blasts Spain in World Cup action"). In developing our Middle East data set, U.S. basketball star Michael Jordan was also a potential source of incorrect codes. Traffic accidents and natural disasters involving multinational fatalities are also a problem, as is transnational criminal activity (unless such activity is being explicitly coded).

Such stories are discarded by using the discard code ###

```
 HEROIN [###]
MARIJUANA [###]
SOCCER [###]
WORLD_CUP [###]
```

If a discard phrase is found anywhere in the source text, no events are coded from the text.

The question of what to discard and what to code depends on the questions that the data are being used to analyze. In the Middle East, activities involving illegal drugs, while certainly present, have relatively little political content. In contrast, when we coded a data set on Colombian and Mexican relations with the United States, illegal drugs were one of the paramount political issues. In those data sets, reports about illegal drugs were not only coded, but a specialized set of agent codes was developed to identify the various entities involved in the drug trade.

KEDS can also detect a number of conditions where a sentence is likely to be too complex to code. For example, the sentence

> Syria said today the U.S. veto of a U.N. Security Council motion on Israeli settlements was ‘‘the most prominent phenomenon of U.S. hostility to the Arabs and U.S. support for Israeli plans to annex the West Bank’’

contains nine actor references to six distinct actors (Syria, U.S., U.N. Security Council, Israel, Arabs, and Palestinians). The actors occur because a complex diplomatic process is being described—for example, the object of the Syrian statement ("U.S. veto of a U.N. SECURITY COUNCIL motion on ISRAELI settlements") involves three actors. Unless the multiple actors are neatly arranged in compound phrases, KEDS usually fails to correctly sort out the subject and object from the modifying phrases.[35] The simplest way to avoid errors such as this problem is to not code sentences that contain an excessive number of actors. Sentences of this level of complexity are not uncommon in Reuters, but in many instance the core event will also be reported in an additional, simpler sentence and therefore coded.

The converse problem occurs when a sentence contains insufficient actors. When coding lead sentences, the absence of an actor before the verb frequently indicates either that the story is a feature that does not involve a political event:

> Kicking up dust, buffalo canter through low brush at this wetland oasis in Jordan’s eastern desert, once a world-famous sanctuary for migrant birds.

A sentence might also contain an excessive number of *verbs*:

> The PLO,raising the stakes beforerenewed Middle East peacetalks, hasaccused the U.S. ofcheating Palestinians byreneging on promises togrant Israel $10 billion inloan guarantees only if ithalted allsettlements in occupied territories.

This admittedly unusual—but authentic—sentence contains seven verb phrases: "raising the stakes," "renewed . . . talks," "accused," "cheating," "reneging on promises," "grant . . . loan," "halt . . . settlements." If the comma-delimited phrase "raising the stakes . . . talks" is removed, KEDS will actually code the sentence correctly because the initial part of the sentence has the SVO structure "PLO accused U.S." But more commonly, multiple verbs are likely to cause coding errors, in part because many of the "verbs" are in fact nouns.

A final problem that might cause a sentence to be too complex to code is intrinsically ambiguous words. For example, "Georgia" can refer either to a region prone to armed ethnic conflict located in the southern part of the former Soviet Union, or a region prone to armed ethnic conflict located in the southern part of the United States. The headline "Bomb blast in Georgia kills two people" does not distinguish the two cases, although in most instances a lead sentence would contain sufficient additional geographical information ("Atlanta" versus "Tblisi") that the location would be clear.

9.7 *Classes and Rules*

KEDS was originally designed as a simple parser working with a small number of grammatical rules and a large number of patterns. As we accumulated experience with the program we encountered a number of situations where a sentence was almost, but not quite, in SVO form, and could be converted to the SVO form by the application of a general pattern-matching rule. We also noticed that general rules could be used to resolve some situations of agent assignment and conjunctions.

[35] In this instance, KEDS would also have incorrectly parsed "Arabs and U.S." as a compound noun, rather than parts of two different compound phrases, leading it to erroneously code ARABS SUPPORT ISRAEL, assuming the program got that far.

While some grammatical rules have been built into the program, KEDS also allows new rules to be specified using two formal grammatical structures, regular patterns and a transformational rules.[36] These facilities are embedded in three features of KEDS:

Classes—These are sets of words that can be used in patterns. These operate in a fashion similar to verbs, actors, agents, pronouns, and other word types recognized by KEDS. Classes typically consist of sets of words that are equivalent either from the standpoint of event coding or in parsing; in a sense classes set up a thesaurus.

Composite patterns—These allow alternative options in a pattern match, as well as patterns within patterns.

Rules—These allow one pattern to be replaced with another pattern.

Classes, composite patterns, and rules are the "bells and whistles" of KEDS—the program works fine (and in fact did for several years) using only simple patterns and some standard English-language syntactic transformations that were coded directly into the source code of the program. However, situations where a number of similar patterns are dealing with the same general syntactical problem can sometimes be solved by creating a rule. General rules using the standard KEDS classes such as <**verb**> or <**agent**> can anticipate situations that are not incorporated into the fixed lists of verb phrases.

The most widely applied rule for coding English is the passive-voice transformation. The safest version of this looks like:

```
 <TOBE> = HAVE_BEEN_ IS_ WAS_ WHEN_ WILL_
<ACTOR1>_<TOBE>_<VERB> BY_<ACTOR2> →
<ACTOR2> <VERB> <ACTOR1>
```

This rule works in most cases, and replaces a large number of patterns of the form

```
- + WAS * BY $
```

However, one potential problem with this formulation is that the actor must come immediately before the <TOBE> verb, which will miss cases where the actor identification is actually an adjective, as in the phrase "Israeli Prime Minister." By eliminating the underscore after <ACTOR1>, the rule can be made more general, but at the risk of making incorrect applications. For example the rule with <ACTOR1> <TOBE> . . . applied to

> Six European countries formally agreed on Thursday to increase the number of observers in the West Bank town of Hebron, most of which was handed over to Palestinian rule by Israel two weeks ago.

reduces the sentence to

[36] In linguistics, the term "transformational" has changed over time. We are using the term in the sense that it was used when Chomsky originally defined categories of grammars: a rule that allows multiple tokens on both sides of a replacement operator. In Chomsky's typology, a KEDS rule is a step more elaborate than the context-free grammars usually encountered in the syntax of computer languages (e.g. Backus-Naur form; Prolog), which only allow multiple tokens on the right-hand-side of the substitution. Context-free grammars, while more familiar in computer science, are insufficient to handle many of the common transformations found in natural language, such as the reversal of subject and object in English language passive-voice.

Six Israel agreed European two weeks ago.

(One way of avoiding this problem would be to restrict the number of words that could occur between parts of a rule; though KEDS does not have such a facility.)

10 Problems with Reuters

All factors considered, Reuters is a remarkably useful source.[37] Using unedited Reuters leads, we have been able to code more than 20 years of some of the most complex political interactions in the world, including the Lebanese civil war and the Arab-Israeli peace negotiations, with a relatively limited vocabulary. This indicates that Reuters maintains fairly consistent levels of editorial control.

Reuters is an international news agency with a large number of readers who are not native speakers of English, so Reuters may use the English language more carefully than, say, the *Washington Post*. Reuters does not, to our knowledge, edit its reports to facilitate machine processing, but such editing is a possibility. International finance and trading firms figure prominently as customers of Reuters and these companies often use computerized filtering systems to automatically route news stories. Nonetheless, four problematic characteristics of Reuters leads should be noted.

10.1 *Attribution*

```
 A Palestinian suspected of collaborating with Israel died Saturday after
being stabbed by masked Arabs the previous day in the occupied Gaza Strip,
hospital officials said.

Iraq said Saturday it did not intend to breach Kuwait's sovereignty but Iraqi
smugglers could be crossing the border to hunt for weapons abandoned in the
Gulf War.

An influential U.S. lawmaker said he was inclined to block further action
on foreign aid this session of Congress, a move that could stall $10 billion
in loan guarantees Israel wants from the United States, the Washington Post
said Friday.

On the eve of a visit by the U.N. General Assembly's president, Israeli army
gunfire killed four Palestinians as the occupied territories erupted in violence
for the second time in four days.
```

The verb SAID, and related verbs such as REPORT, are used in two different ways in Reuters. In the first example, an entity that is not an actor is reporting an event. In the second, a political actor is making an official statement. The third example contains both of these uses of SAID in the same sentence. The final example has no attribution. All four types of attribution are found in Reuters leads, although the fourth (no attribution in the lead) is the most common.

The KEDS coding system treats the isolated verb SAID as subordinate, which means it is coded (as a comment, WEIS 02) only if no other event is found in the lead. Thus SAID will be ignored in

[37] Because almost all of our coding experience has been with Reuters, this section specifically addresses that source. If Reuters continues to remain inaccessible to the academic community, it is likely that most event data projects will switch to an alternative source such as AFP, but most of the problems we discuss here will be found in any newswire source.

the first case (STABBED is the verb) but coded in the second, which contains no other verb in our dictionary. In the third sentence, however, the SAID is the principal verb but would be ignored if another verb phrase, such as "block . . . aid" was picked up.

SAID, combined with a pattern giving a direct object, is often the primary verb of the sentence, and SAID has the longest pattern list of any of the verbs in our dictionary. This is partly a function of the WEIS coding scheme, which distinguishes between a variety of verbal behaviors, but is also a function of the fact that a great deal of international interaction is verbal behavior. SAID is also an unusual verb for KEDS because it is used as a coded and a subordinate verb with almost equal frequency.

The subordinate SAID also means that when one actor is making a statement concerning activities by two other actors, this will be coded as if the event actually occurred:

> *Lebanese* Prime Minister Rashid Karami said today the *Soviet Union* had agreed for the first time to help finance the *U.N.* peacekeeping force in Lebanon.

While Reuters tends to use verbs such as ACCUSED or ALLEGED in these circumstances, and reserves SAID for sources Reuters considers authoritative, KEDS codes a certain number of events from sentences that are actually just comments. The number of such cases is probably small but they do add noise to the data.

Another dimension of the attribution problem is the embedding level. Consider again the complex sentence given in section 2.6.6:

> *Syria* said today the *U.S.* veto of a *U.N. Security Council* motion on *Israeli* settlements was ''the most prominent phenomenon of *U.S.* hostility to the *Arabs* and *U.S.* support for *Israeli* plans to annex the *West Bank.*''

There are at least four levels of attribution here:

Primary:	*U.S.* veto of a *U.N. Security Council* motion
Secondary:	*U.S.* veto of a *U.N. Security Council* motion on *Israeli* settlements
Tertiary:	*Syria* said today the *U.S.* veto of a *U.N. Security Council* motion on *Israeli* settlements was "the most prominent phenomenon of *U.S.* hostility to the *Arabs*
4th level:	*Syria* said today the *U.S.* veto of a *U.N. Security Council* motion on *Israeli* settlements was the most prominent phenomenon of . . . *U.S.* support for *Israeli* plans to annex the *West Bank*"

By the final level, we are dealing with the Syrian interpretation of how a U.S. action on a U.N. action concerning an Israeli action reflects U.S. policy support for a related putative Israeli policy. While sentences containing fourth-level references are relatively uncommon in Reuters leads, tertiary references are quite common.

One can question whether statements such as this should even be in an event data set, irrespective of whether they can be coded correctly. Our sense, based on reading a variety of Reuters reports, is that secondary attributions that are policy statements—X commenting on Y's actions toward Z—carry some information, but not very much. When an important policy statement is being made, usually one will find a sentence reporting the primary event—Syria denounced the U.S.

veto of U.N. . . . —that can be coded. Furthermore, in some instances events involving attribution are a direct result of Reuters or other international media seeking comments or interviews rather than spontaneously generated events.

While we have not systematically assessed the impact of attributed statements, our sense is that this could be eliminated without having a lot of effect on the event stream. McClelland [23] noted that the comment event category in WEIS had been added as an afterthought, despite the fact that it accounts for about a third of the events in the ICPSR WEIS data set. This category has the lowest weightings in the Goldstein [11] and Azar and Sloan [4] scales, and thus has little effect in analyses that use aggregated scaled data. The coding of comments is further complicated that the fact that distinguishing the positive or negative affective component of a comment involves a great deal of human judgment about the context of the statement, which in turn makes the coding prone to error and inconsistencies. Disregarding attributed statements—and certainly avoiding anything more complex than a secondary attribution—would probably result in a cleaner data set.

10.2 *Unidentified Source*

```
 A suicide car bomber has killed 12 Israeli soldiers and wounded 14 just across
the Lebanese border in the most lethal attack against Israeli forces since
the start of their withdrawal from South Lebanon.

An Arab shot and killed by two men in downtown Athens was identified today
as a top-ranking member of the Palestine Liberation Organization, the group
said today.
```

This type of lead is relatively common in areas where there has been a breakdown of political authority. In such circumstances, many violent interactions are reported that involve "unnamed gunmen," "guerrilla groups" and other anonymous agents, and unaffiliated violent events occur ("A bomb exploded near a checkpoint . . . "). This problem is not generated by Reuters but by the nature of the political action.

In a protracted conflict, these anonymous events probably have little effect on the overall event data series, since ample events occur where the identities of the actors are known. In addition, a group (or frequently, multiple groups) will usually claim responsibility for an event after the fact, and these claims generate events.

10.3 *Specialized Vocabulary*

The Reuters vocabulary is generally quite stable. However, in generating our Middle East data sets, one of the most difficult periods to code was the 1990-91 Iraq-Kuwait crisis, particularly the military operations. This period required substantial new vocabulary development because it dealt with a classical international invasion and a multi-lateral military response, behaviors we had not encountered before in coding a decade of low-intensity conflict in Lebanon and Palestine. During military operations, for example, the ambiguous verbs ATTACK and FORCE usually refer to physical rather than rhetorical actions. One way to deal with this problem might have been to develop specialized dictionaries to deal only with this period, and then revert to our standard dictionaries once the crisis had ended.

10.4 *Feature Stories*

While the majority of the reports in Reuters refer to political events, there are a number of feature stories providing human interest, political analysis, or historical background. For example, 15 of the 20 leads on 1 January 91 deal with meetings, warnings, political comments, troop and refugee movements, and other activities that can be coded as events, but five do not:

> New Year's Day moved the countdown to the U.N. deadline for Iraqi forces to leave Kuwait into its final fortnight, with no one optimistic that last-ditch diplomatic efforts would avert a major war.
>
> U.S.-led forces are likely to take up to three days, rather than hours, to gain air supremacy over Iraq if war erupts in the Persian Gulf, Western military sources say.
>
> Britain was worried that Iraq might attack Kuwait 30 years ago and drew up plans to dislodge Iraqi troops from the territory, according to cabinet papers.
>
> An African witch doctor who divines the future with magical stones says there will be a short war in the Persian Gulf which will see limited loss of life and the defeat of Iraq.
>
> ''Storming'' Norman Schwarzkopf, commander of half a million Americans poised for war against Iraq, is a tough guy in public whose idea of relaxation is to listen to wild ducks squawk on tape.

The first two leads are analysis, the third is historical, and the final two presumably are for human interest.[38]

In general, leads for feature stories do not contain verbs corresponding to events and are consequently not coded. Occasionally, however, they do, particularly when dealing with historical events. For example the third story in this example contains the SVO sequence "Britain attack Kuwait." In addition, KEDS does not deal with conditional or hypothetical statements, so scattered throughout the machine-coded sequences are an assortment of non-existent acts of force among improbable dyads. These occur at random and would never be confused with the sustained conflict found in Lebanon, Desert Storm or the *intifada*, but any analytical techniques that were highly sensitive to unanticipated uses of force would need to filter them, much as U.S. nuclear warning systems learned to ignore the occasional flocks of geese and software glitches that computers mistook for incoming Soviet ICBMs.

[38] Although one cannot help but notice that the African witch doctor's predictions were substantially closer to the actual crisis outcome than those being made by many more conventional analysts at the time. Over the next three days, for example, we find:
-Yasser Arafat said in a newspaper interview published Wednesday no one would dare unleash a Gulf attack and predicted a normal day when a U.N. ultimatum expires in two weeks. (2 Jan)
-Luxembourg, hosting European Community talks on the Persian Gulf crisis, predicted that Iraq would pull its troops out of Kuwait at the last minute in order to avoid war. (2 Jan)
-President Bush said he would like Congress to approve possible use of force in the Persian Gulf but was told the Senate was unlikely to support such a request, Senate Democratic leader George Mitchell said. (3 Jan)
-War in the Persian Gulf threatens to turn a recession in Britain into a deep slump, jeopardizing Prime Minister John Major's election prospects, economists say. (4 Jan)
-From the ever-resourceful Reuters we also learn that "astrologers in the Himalayan kingdom of Nepal" (16 Jan) got the outcome correct; Philippine psychics (8 Jan) did not. Western analysts, of course, hedged: "Western defense experts are predicting any war to dislodge Iraq from conquered Kuwait may last three days to a month" (10 Jan).

11 Discussion

As Weber [39, pg. 17] notes, reliability in a content analysis project consists of three components:

stability—the ability of a coder to consistently assign the same code to a given text;

reproducibility—intercoder reliability;

accuracy—the ability of a group of coders to conform to a standard.

For a given set of patterns, the stability of machine coding is 100% because the machine will always code the same text in the same manner. This is particularly useful when a time series is being maintained for a number of years. Because the patterns used in coding are explicitly specified in the coding dictionaries rather than dependent on coder training, the same rules used in 1992 can be used in 2002. In our experience, the reproducibility of machine coding seems comparable to the inter-coder reliability of humans, and a machine is obviously not influenced by the context of an event or by intrinsic political or cultural biases. (The coding dictionaries may reflect biases, but these will be explicit and can be examined by another researcher; the dictionaries are also applied consistently to all actors and in all contexts.) Furthermore, the machine is not subject to coding errors due to fatigue or boredom, and, once a coding vocabulary has been developed, it does not require retraining.

KEDS was developed inductively. Because international event data are based on an SVO framework and most declarative sentences in English have that form, a few simple rules can be used to generate a large number of correctly coded events without human intervention. In the early versions of the program, when the dictionaries were very limited, it was also important to use shortcuts such as stemming to extract information in a syntactically incorrect fashion, such as interpreting nouns and adjectives as if they were verbs. At that stage of the KEDS project, we were also working in two languages (English and German) and were thus reluctant to incorporate features specific to English.

As our coding dictionaries became more elaborate (and as we added agents), the nature of the coding errors changed. Most verbs are now coded correctly and a higher proportion of the errors are due to incorrect source and target identification. These errors are caused by complex sentence constructions, subordinate and adjectival clauses, and ambiguous pronouns, and therefore cannot be solved by simply adding vocabulary.

KEDS is thus facing something of a dilemma. A few of the remaining problems can be solved by adding specific grammatical features, and we are experimenting with several of these using KEDS's rules facilities. Beyond this level, however, one needs to develop a much more sophisticated parser because most of the Reuters leads that create coding problems are quite syntactically complex.

There are, however, at least three arguments against developing a more sophisticated parser. First, sentences too complex to correctly code using the KEDS approach probably account for fewer than 10% of all Reuters leads. KEDS's sparse parsing is relatively robust in the sense that many complex sentences are assigned the correct code for the incorrect reason. An additional 5% to 10% of the sentences are those where even human coders will disagree, either because of ambiguities in the report or, more commonly, ambiguities in event coding schemes such as WEIS.

Second, we are not aware of any "off-the-shelf" parser-coders that perform substantially better than 90% on unedited source text, though this could change in the future as the capacity of computers and the complexity of parsing models continues to increase. At present, there are numerous parsers and natural language understanding systems that are capable of doing much more sophisticated coding and classification than KEDS in *limited* behavioral domains and/or with *pre-edited* text, but that is not the problem we need to solve. KEDS is designed to take *unedited* sentences from Reuters and other news services covering a very *wide range* of political behavior.

Finally, we suspect that a more sophisticated approach will probably involve semantic information—

information pertaining to word meanings rather than simply their type and relationship to each other—and this will involve a very substantial amount of work in terms of dictionary development. For problems more complex than event data coding, parsers with semantic information are often essential, but we suspect they are not needed for our problem.

11.1 *Future Directions*

The previous discussion is not intended to imply that an improved machine-coding system could not be created; in fact we would be absolutely delighted to see such an effort undertaken. It is simply to say that such a project will probably involve a substantial amount of work and from the standpoint of *our* research, it is more important to concentrate on refining the event coding schemes and the methods of analyzing event data rather than squeezing out another 5% accuracy, particularly since that additional information may make very little difference in the statistical results obtained from the data.

If such a project were undertaken, however, we would suggest focusing on the following problems:

11.2 Attribution

Systematically dealing with the attribution problem could affect the coding of a very large number of sentences in any set of news wire or newspaper texts. However, this is at least as much an issue of refining the definitions in the event coding scheme as it is in machine-coding per se.

11.3 Disambiguation of Nouns and Verbs

Coding would be improved in a system that distinguishes whether words are being used as nouns or verbs. This would involve a fairly complete parsing of the sentence—and it would certainly require tagging almost all words in the sentence with the appropriate part-of-speech (including adjectives, adverbs, articles, and prepositions, which KEDS ignores unless they are required to distinguish the coding of a verb). However, there are several very extensive dictionaries available in the public domain (for a general list, see http://www.facstaff.bucknell.edu/rbeard/ diction3.html that provide this information, and contemporary computers (unlike the computers on which we initially developed KEDS) have the memory and processing power to handle this easily.

11.4 Automatic Identification of Nouns That Are Not in the Actor List

This would eliminate incorrect codings that result from KEDS only "seeing" the nouns it expects to see, and therefore sometimes assigning an incorrect subject or object. For example in the sentence

```
Undercover police on Wednesday arrested two clowns who peddled drugs to young
people in Mexico City's Chapultepec park, the attorney general's office said.
```

the direct object CLOWNS—whether conventional or drug-peddling—is unlikely to be in most agent lists, and a typical coding system would instead pick up POLICE ... ARRESTED ... PEOPLE. A more sophisticated system would be able to recognize that CLOWNS was the object of ARRESTED, and therefore bypass the erroneous coding of PEOPLE as the object, even if the system didn't know what to do with the word.

11.5 More Effective Use of Sets of Synonyms and Verb Forms

KEDS has the capability of handling synonyms through its CLASSES facilities, but that was added to the program at the end of its development rather than incorporated into the dictionary structure from the beginning.

For reasons discussed earlier, KEDS dictionaries are organized by actors and verb phrases, rather than by codes. While this makes sense from a *parsing* perspective, from a *coding* perspective, organizing by code probably makes more sense—in other words, any event code has an acceptable set of verbs and an acceptable set of objects. For example, a subset of the code for "give financial aid" might look like

Verbs: [promise, pledge, grant, allocate, award]
Objects: [dollars, DM, yen, ecus, financial_aid, financial_assistance, bailout, debt_relief]

The obvious advantage of this scheme is that it is equivalent to 40 verb phrases. A secondary advantage is that it might also result in more transparent and easily-modified coding schemes, which could use existing lists of equivalent verbs and objects. Coding dictionaries could be organized around these sets of equivalent words rather than around individual words. In some cases these synonyms will be based on natural language usage—the *WordNet* synonym sets (http://www.cogsci.princeton.edu/∼wn/) are a particularly important resource here—and in other cases the sets will be specific to a type of behavior being coded.

11.6 Filtering Based on Temporal Relationships

Reuters leads frequently deal with events in the near future or the recent past; the program does very little to distinguish these. To the extent that one can depend on a news agency weighting of the importance of various events by the extent of its coverage of those events, this can actually be useful: An activity that a news agency deems important—for example, a meeting between chief executives or between antagonists—will generate many more events than an activity that is routine, such as a meeting between agricultural ministers of two allies. However, a cleaner event sequence would be generated by coding only events that occur in the present, and this would also eliminate some of the widely miscoded events that result from reports of historical events and anniversaries.

11.7 Determining Pronoun References Across Sentences

A major impediment to coding full stories is the ability to handle pronoun references to earlier sentences

```
 Islamic foreign ministers meeting in Burkina Faso have prepared a draft
resolution condemning Israel for its blitz of Lebanon last week and demanding
that it pay reparations, officials said.

On Iraq, they call for a lifting of sanctions in return for Iraqi respect
of U.N. Security Council resolutions.

On Kashmir, they offer support to the region's search for autonomy.

On Kosovo, they demand an accelerated return of refugees and reconstruction
of the southern Yugoslav province.
```

Based on our work on the Middle East, we originally thought this would be relatively easy to solve, since pronouns in those stories usually refer to the first actor in the previous sentence, as in these examples.[39] However, this seems to be a quirk of some unknown (but we love 'em!) Reuters editor handling the Levant, and this technique has not worked as well in other regions of the world. Determining pronoun references *across* sentences is more difficult than determining the references *within* sentences because the previous sentence may contain a number of different actors. As noted earlier, at times the problem is simply unsolvable. Nonetheless there are a number of methods available in the NLP literature that could be applied to it. Alternatively, one might simply set some additional complexity conditions to filter out sentences that are likely to involve cross-sentence pronoun references.

11.8 Full Parsing

Despite the arguments presented above for continuing with sparse parsing, natural language processing technology may have reached a point where a coder based on full parsing is appropriate. Full parsers are far better than sparse parsers at handling disambiguation, and can also deal effectively with issues such as subordinate phrases, negation, and the different forms of verbs. Fortunately for our efforts, parsing is a very general problem that occurs in a wide variety of contexts, including automated indexing and retrieval, database query, speech recognition, and machine translation.

The development of parsers for English and other languages has been an active research area in computational linguistics for at least three decades. This would be consistent with the broader incorporation of automated text processing into political science [25] 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`
- 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.

However, most existing automated content analysis systems developed in the twentieth century do not use parsing. For example, a recent review of 15 contemporary software packages for computer-assisted text analysis by ZUMA (originators of the widely-used TEXTPACK system) found that only three "incorporate linguistic information" (and one of those was KEDS) ([1]:147).

A "robust parser"—one designed to work with unedited text in general subject domains—can also deal with a variety of other tasks. For example, Carnegie Mellon's "LINK Parser" (`http://www.link.cs.cmu.edu/link/`)[40]

[39] The occurrence of the multiple "they" reference, on the other hand, is quite unusual in Middle East leads, though it may be common for the Reuters reporter in Ouagadougou, where this story originated.

[40] A link [sic] that is quite remarkably still valid after more than a dozen years.

> . . . has a dictionary of about 60000 word forms. It has coverage of a wide variety of syntactic constructions, including many rare and idiomatic ones. The parser is robust; it is able to skip over portions of the sentence that it cannot understand, and assign some structure to the rest of the sentence. It is able to handle unknown vocabulary, and make intelligent guesses from context about the syntactic categories of unknown words. It has knowledge of capitalization, numerical expressions, and a variety of punctuation symbols.

With machine parsing, the parser would first "mark-up" the sentence, and then the coder would operate on the marked-up version, rather than from the original text. In principle, a user could therefore substitute an entirely different parser—for example, a commercial parser, a language-specific parser, or a domain-specific parser—without modifying the coding model.

The disadvantage of the full-parsing approach is that the dictionaries would need to be substantially more complex in order to use the additional information. At the present time, there appears to be no single standard for marking syntactic structures, or even parts of speech, so the dictionaries would in fact be linked to a specific parser. Furthermore, because basic event coding requires knowing only the subject, verb and object of a sentence, most of the effort involved in full parsing will be wasted. Some of these parsers are very slow, and the processing time could become a significant constraint in a project involving a large amount of text.

11.9 *A Final Note on the Psychopathology of Dictionary Tweaking*

When are machine-coded data "good enough"? This is a very difficult issue, and some people appear to have a profound psychological block to using any data set that might, with additional tweaking of the dictionaries, be further improved. Yet if one backs off and looks at the overall research exercise, it is abundantly clear that a research project can very quickly reach a point of diminishing returns in dictionary development. As we will discuss in the next chapter, event data are produced from a nonrandomly censored sample of source texts, categorized with an imperfect set of codes, arbitrarily scaled into interval-level variables, and then applied in mis-specified models.

We suspect that the attractiveness of dictionary tweaking comes from the *illusion* of control over the analytical process—add a verb phrase, and you can see an event go from a use of force to an innocuous verbal protest. The data are visibly better—progress! Let's grab another cup of coffee and keep on tweaking![41]

Don't kid yourself. The phrase that was just changed may occur once or twice in the entire set of texts, and there's a non-zero chance that the addition will mess up other phrases that were previously coding texts correctly (we speak from experience). Even the addition of hundreds of phrases, will probably changing less than 5% of the coded events. If the Actor_Filter program was used, you already know that no major actor references have gone unnoticed. Tens of hours of tweaking may result in only negligible (and random) changes in the coefficient estimates of the final model.

Except for the addition of vocabulary to deal with regionally-specific activities such as drug wars and collapsed Ponzi schemes, the time and effort that one is tempted to spend on dictionary tweaking is usually far better spent on refinement of statistical models and estimation techniques. This can be a much less satisfying process: Instead of quietly punching the <Return> key and periodically solving little word puzzles, one must struggle to comprehend esoteric matrix notation, cope with partially-debugged algorithms borrowed from econometricians, deal with the unsatisifying tradeoffs between specification error and collinearity, and make unknowable decisions about error

[41] "Hi, my name is Jim and I tweak dictionaries"?

covariance structures. You have to try numerous model specifications that go nowhere, and when a model specification finally gives plausible results, you wonder whether you've merely out-foxed the significance test.

Coding is easy. Data analysis is hard.

Okay, we're over-stating the case here to make a point. We are certainly delighted to see additional work done with the KEDS dictionaries, and in some circumstances—particularly those where the lines between internal and international events are very murky, as in the Balkans or the Great Lakes region of central Africa—our standard dictionaries may require substantial augmentation. The same is true if one is coding internal events or activities that go beyond the Westphalian framework of the WEIS system, such as economic interactions, criminal activity, environmentally induced stress or refugee movements. In these cases, it is probably a very good idea to develop a new coding system in addition to working on new dictionaries. WEIS wasn't intended to cover all circumstances, and it doesn't.

But there are limits. During the first decades of the automobile, a popular cultural motif was the picture of a car being dragged out of the mud by a horse. You'all got this fancy technology, but in the end you're still gonna need my horse. But a horse can't travel 60 mph for ten hours a day, and a human can't code 45 events per second. Human coders, meanwhile, are an amalgam of biases, misperceptions, sloth, confusion and neglect who graduate and vanish at the end of the year (or earlier). Anyone can find examples of stories that automated systems badly miscode—the KEDS and PANDA projects have archives full of our favorite errors. But those errors are embedded in tens of thousands of correctly coded events and, with proper statistical treatment, these are nearly irrelevant in any well-designed statistical analysis.

A common criticism of the DARPA studies of the 1970s was that they did beer-budget analysis on champagne-budget data. If a research project can't cope with the fact that at least 15% of the events—whether human or machine coded—are in error, the analysis should not be done with event data. But just as projects using survey data have learned to deal with sampling and nonresponse biases using statistical techniques, so can event data analysts learn to deal with the errors in our form of data. As the studies later in this volume will demonstrate, event data isn't perfect, but it's good enough.

References

[1] Melina Alexa and Cornelia Zuell. *A Review of Software for Text Analysis*. Zentrum für Umfragen, Methoden and Analysen, Mannheim, 1999.

[2] Hayward R. Alker. Emancipatory empiricism: Toward the renewal of empirical peace research. In Peter Wallensteen, editor, *Peace Research: Achievements and Challenges*. Westview Press, Boulder, CO, 1988.

[3] Advanced Research Projects Agency (ARPA). *Proceedings of the Fifth Message Understanding Conference (MUC-5)*. Morgan Kaufmann, Los Altos, CA, 1993.

[4] Edward E. Azar and Thomas Sloan. *Dimensions of Interaction*. University Center for International Studies, University of Pittsburgh, Pittsburgh, 1975.

[5] Roy F. Baumeister and John Tierney. *Willpower: Rediscovering the Greatest Human Strength*. Penguin, New York, 2011.

[6] Doug Bond, Brad Bennett, and William Vogele. Data development and interaction events analysis using keds/panda: an interim report. Paper presented at the International Studies Association, Washington, 1994.

[7] Doug Bond, J. Craig Jenkins, Charles L. Taylor Taylor, and Kurt Schock. Mapping mass political conflict and civil society: Issues and prospects for the automated development of event data. *Journal of Conflict Resolution*, 41(4):553–579, 1997.

[8] Philip M. Burgess and Raymond W. Lawton. *Indicators of International Behavior: An Assessment of Events Data Research.* Sage Publications, Beverly Hills, 1972.

[9] DARPA. *DARPA Message Understanding Proceedings 1991.* Muc-3 : Proceedings of a Conference Held in San Diego, California, May 21-23, 1991. Morgan Kaufmann, 1991.

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

[11] Joshua S. Goldstein. A conflict-cooperation scale for WEIS events data. *Journal of Conflict Resolution*, 36:369–385, 1992.

[12] Joshua S. Goldstein and Jon C. Pevehouse. Reciprocity, bullying and international cooperation: A time-series analysis of the Bosnia conflict. *American Political Science Review*, 91(3):515–530, 1997.

[13] Random House. *The Random House College Dictionary, rev ed.* Random House, New York, 1975.

[14] Valerie Hudson, editor. *Artificial Intelligence and International Politics.* Westview, Boulder, 1991.

[15] Phillip A. Huxtable. *Uncertainty and Foreign Policy-making: Conflict and Cooperation in West Africa.* Ph.d. thesis, University of Kansas, 1997.

[16] Robert Jervis. *Perception and Misperception in International Politics.* Princeton University Press, Princeton, 1976.

[17] Daniel Kahneman. *Thinking Fast and Slow.* Farrar, Straus and Giroux, New York, 2011.

[18] Yuen Foong Khong. *Analogies at War.* Princeton University Press, Princeton, 1992.

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

[20] Edward J. Laurance. Events data and policy analysis. *Policy Sciences*, 23:111–132, 1990.

[21] Richard Ned Lebow. *Between Peace and War: The Nature of International Crises.* Johns Hopkins University Press, Baltimore, 1981.

[22] Stan Malless and Jeff McQuain. *The Elements of English: A glossary of basic terms for literature, compositions and grammar.* Madison Books, New York, 1988.

[23] Charles A. McClelland. Let the user beware. *International Studies Quarterly*, 27(2):169–177, 1983.

[24] Slava Mikhaylov, Michael Laver, and Kenneth Benoit. Coder reliability and misclassification in the human coding of party manifestos. *Political Analysis*, 20(1):78–91, 2012.

[25] Burt Monroe and Philip A. Schrodt. Editors' introduction: The statistical analysis of political text. *Political Analysis*, 16(4), 2008.

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

[27] Steven Pinker. *The Language Instinct*. W. Morrow and Co., New York, 1994.

[28] Andrea Ruggeri, Theodora-Ismene Gizelis, and Han Dorussen. Events data as bismarck's sausages? intercoder reliability, coders' selection, and data quality. *International Interactions*, 37(1):340–361, 2011.

[29] Philip A. Schrodt. Automated production of high-volume, near-real-time political event data. Presented at the American Political Science Association meetings, Washington, 2010.

[30] Philip A. Schrodt and Deborah J. Gerner. Statistical patterns in a dense event data set for the middle east, 1979-1992. Presented at the Midwest Political Science Association, Chicago, 1993.

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

[32] Philip A. Schrodt and Deborah J. Gerner. Empirical indicators of crisis phase in the Middle East, 1979-1995. *Journal of Conflict Resolution*, 25(4):803–817, 1997.

[33] Eric Singer and Valerie M. Hudson, editors. *Political Psychology and Foreign Policy*. Westview Press, Boulder, 1992.

[34] Philip E. Tetlock. *Expert Political Judgment: How Good Is It? How Can We Know?* Princeton University Press, Princeton, NJ, 2005.

[35] G. Dale Thomas. *The "Strange Attractiveness" of Protracted Social Conflict in Northern Ireland*. PhD thesis, University of South Carolina, 1999.

[36] Rodney G. Tomlinson. World event/interaction survey (WEIS) coding manual. Mimeo, Department of Political Science, United States Naval Academy, Annapolis, MD, 1993.

[37] Yaacov Y.I. Vertzberger. *The World in their Minds: Information Processing, Cognition and Perception in Foreign Policy Decision Making*. Stanford University Press, Stanford, 1990.

[38] Jack E. Vincent. WEIS vs. COPDAB: Correspondence problems". *International Studies Quarterly*, 27:160–169, 1983.

[39] Robert Philip Weber. *Basic Content Analysis*. Sage Publications, Newbury Park, CA, 2d ed. edition, 1990.