Journal of
Peace Research

Journal of Peace Research
2015, Vol. 52(1) 125–128
© The Author(s) 2014
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0022343314523807
jpr.sagepub.com
SAGE

Making uncertainty explicit: Separating reports and events in the coding of violence and contention

Nils B Weidmann

Department of Politics and Public Administration, University of Konstanz

Espen Geelmuyden Rød

University of Konstanz & Conditions of Violence and Peace Department, Peace Research Institute Oslo (PRIO)

Abstract

When coding events from media sources – as the majority of data projects do – different reports may oftentimes contain contradictory information. What do coders make of this? It is up to them to aggregate different reports into one coded event, and to supplement missing information based on other sources or their own background information. If not addressed properly, this may lead to a lack of replicability and to low reliability of the final data product. In this short article, we present an approach for separating (i) event *reports* and the information contained in them, and (ii) *events*, which are based on aggregate information from the reports and constitute the final data product. Our procedure preserves uncertainty arising from multiple reports and gives the user control over how missing and conflicting information should be dealt with. We illustrate our procedure with data from a current coding project, the Mass Mobilization in Autocracies Database (MMAD).

Keywords

event data, media bias, protest

Motivation

The majority of event coding projects on political contention and violence rely on media reports. As Salehyan describes in the introduction to this special section, the coding process typically starts from a set of ‘primary reporting’ sources to the ‘information extraction’ stage, in which coders retrieve information about the phenomenon of interest from these sources, and transform the information into the format required for the final data product (in our case, events). Extracting and transforming information from sources, however, is much more complex than it seems. First, there is the challenge of isolating the relevant bits of information from a news report. Which parts of the report should be used to determine the nature of an event? What locational information is there to tell us where it happened? Second, it is sometimes difficult to bring extracted information into the format that is required for the final database: oftentimes, there will be multiple reports from different sources about a single event, and coders will have to aggregate them into a single entry.

So far, we lack sufficiently transparent procedures to deal with these challenges. With little specification of how information extraction and aggregation should be done, it is left to the coder to find the relevant pieces of information across a set of reports and aggregate them to a set of events. In other words, the process of getting from a news report to an event coding remains a black box, which is unsatisfactory for two reasons. First, it severely hampers a user’s understanding of how the data were generated, thus reducing the transparency of the

Corresponding author:
nils.weidmann@uni-konstanz.de

coding process. If, for example, a protest event mentions a certain number of protesters, how can we know if the number comes from a single source, or if multiple sources report similar (or potentially different) numbers? Second, it makes replication of a particular coding effort essentially impossible. With various efforts to improve not only statistical, but also data replication in the discipline, a more transparent design of the coding process is necessary.

This short article is a first attempt at dealing with these challenges. We illustrate a simple procedure that addresses the problems mentioned above by separating the process of information extraction from the process of information aggregation. To be sure, our discussion is primarily relevant for high-resolution event datasets, which usually rely on multiple media sources. Furthermore, although we illustrate our procedure using protest event data, we believe that the insights provided can be of use for any event data collection effort using multiple sources.

From news reports to events

The coding process translates raw information into a simplified, often numeric representation of events that allows for large-N analysis. In many cases, this raw information comes from news reports, but alternative sources such nongovernmental organizations or government agencies are also used. We refer to this source information as 'reports'. Reports rarely come in a form convenient for coding. Instead, two steps need to be performed during the coding: first, the relevant pieces of information need to be extracted from the report. For example, this information can be about location and time of an incident, about the issue of a protest, and about the number and type of participants. This is the 'information extraction step'.[1] Second, once the relevant information has been determined across a set of reports, it needs to be aggregated into a set of events, which constitute the final dataset. Oftentimes, the coding of one event is based on more than one report, or different single-day reports are combined into a longer event. This is the 'aggregation step'.

Figure 1 provides a stylized illustration of the coding process. For now, consider only the two dashed boxes on the left and right. The left box shows a set of two reports, which together constitute the source material for a coding project. For example, this can be a collection of news articles obtained from a news archive such as Lexis-Nexis or Factiva. The box on the right shows the final dataset, which in our case is a set of individual events. Usually, data projects do not specify how exactly they get from the source material to the final dataset. While almost all datasets specify the type of information that needs to be available before an event is coded, the fact that both information extraction and aggregation are performed by the coder without precise guidelines leaves us with two problems: First, we do not know the precise formulation of certain types of information in a report. For example, was the location reported as a precise city, or as a village outside a city? What was the precise label the news report used for a group of protesters? Without transparent coding rules, it is essentially up to the coder to map a particular piece of information in a report to the corresponding entity (i.e. a location or a group) that the dataset relies on. A similar but probably even more severe problem arises when the coder aggregates different pieces of information. For example, if two reports are about the same event but mention different numbers of protesters, which one was used in the final coding? How do we know that multiple, rather than one, sources were used to generate the event?

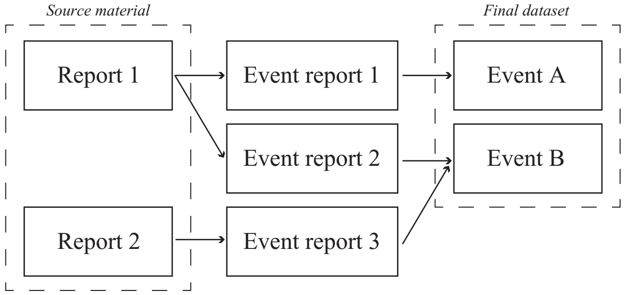

Figure 1. Event reports

In order to remedy the problems mentioned above, we propose to separate the information extraction and the aggregation steps. Essentially, the idea is to introduce an intermediate type of output from the coding process, which we call 'event report'. As the name suggests, an event report is an individual statement of an event, derived from a news report. It contains fields for all the relevant information we need to eventually code an event. Thus, an event report is the output of the information extraction step, which serves as input to the aggregation step to generate the final list of events. Figure 1 illustrates this. From the source reports (left) we extract a set of event reports (center), which are later aggregated into the events that constitute the final dataset (right).

[1] We use the term 'information extraction' in a more narrow sense than Salehyan (2015, this issue) to refer only to the task of determining relevant parts of a report.

While many reports will only contain information about a single event (for example, Report 2 generates only one event report, no. 3), this procedure is able to deal with more complex reports: Report 1 mentions two events, which result in Event reports 1 and 2. Once we have generated the set of event reports, we need to aggregate them to obtain the final dataset. Again, many events will be based on only one event report, as is the case for Event A in Figure 1. However, in cases where there are multiple event reports for one event, the coder will have to aggregate them into one event. Since the extracted information is provided in a standardized form in the event record, this process can largely be automated, thus making it extremely cheap and transparent. We will provide an example for this below.

How does this procedure solve the problems we discussed above? First, it makes the information extraction step much more transparent. Using the event report(s) that an event is based on, a user can find out, for example, what phrase in the report was used to pinpoint the location of an event. This applies to other types of information, such as the number and type of protesters, or the issue of the protest. Also, the user has full information about, and can even control, the aggregation process. For example, it is possible to change the way that participant numbers from the event reports are aggregated into a single number, or even to weigh information by source. Last, the event records can serve as training data for automatic text coding of event data. To date, these routines perform information extraction and aggregation in a single step, similar to human coding. This leads to exactly the same concerns we described above, in particular regarding information aggregation. In contrast, using the intermediate stage of event reports, we can develop computational routines for information extraction (i.e. the creation of event reports) and the aggregation of these reports, and thus improve transparency of automated coding techniques.

Our procedure adds a new type of output to the coding process: the list of event reports. These event reports would have to be distributed alongside the finished list of events. In addition, users need to be provided with the aggregation routine that moves from event reports to events.[2] Only then will it be possible for users to systematically trace and modify the data generation process that results in the finished event dataset. We believe that the advantages of this process greatly outweigh the complexities it entails. Today's event datasets often shield much of the complexity of the coding process from the end user, making the final event list look complete and accurate. As probably all creators of these datasets will admit, this is simply not true.

Table I. Event reports for Osh (Kyrgyzstan), 21 March 2005

Number of participants	*Security forces engagement*	*Source*
hundreds		AFP
2,000	present (1)	AP
1,000		AP
	present (1)	AP
	present (1)	BBCM
1,000	not present (0)	BBCM
several thousand		BBCM
3,000	physical intervention (2)	BBCM
200		BBCM

Example from the MMAD project

We illustrate the use of event reports using the Mass Mobilization in Autocracies Database (MMAD), which contains events of protest in autocracies with precise spatial and temporal coordinates for the years 2003–12. The coding process uses the above-mentioned 'event reports' as intermediate coding products, and these are are later aggregated to individual events. In this section, we illustrate the problem of contradictory information in the chosen sources using protests during the Tulip Revolution in Kyrgyzstan. We describe how information from different event reports can be aggregated into a single event coding, according to different aggregation rules.

Table I displays nine event reports with divergent information on two variables included in MMAD, the number of participants and the level of security forces engagement (ordinal, values in parentheses).[3] The last column displays the news source.[4] All of the event reports in Table I took place in the city of Osh on 21 March 2005. It is immediately apparent that the information in the event reports diverges both across reports from the same source and also across sources: the number of participants differs in all three reports from AP and all five reports from BBCM. Also, the estimate given by AFP (hundreds) is very different from the AP estimates of 1,000 and 2,000 and three of the BBCM estimates (1,000, several thousand, and 3,000). In addition, there

[2] Using established data management tools such as relational databases and SQL, this is straightforward.

[3] Other variables omitted for the sake of illustration.

[4] The sources used for coding event reports in MMAD are BBC Monitoring (BBCM), Agence France Presse (AFP), and Associated Press (AP). These sources were chosen based on a source selection process described in detail in Rød & Weidmann (2013).

Table II. Alternative event codings for Osh (21 March 2005), according to two different aggregation rules

Aggegation rule	*Number of participants*	*Security forces engagement*
average(no. of participants), mode(security forces engagement)	1,440	present (1)
max(no. of participants), max(security forces engagement)	3,000	physical intervention (2)

are two reports without participant number estimates. There is similar uncertainty regarding the security forces' involvement in the protest. In fact, the information ranges from not present (0) to physical intervention (2). Four event reports indicate presence or intervention, four do not mention security forces' involvement at all, and one asserts no presence. Absent any transparent guidelines, it is not clear how different coders would have aggregated these event reports in a conventional event dataset.

Once we have extracted these event reports from the selected news reports, we need to aggregate them up to the level of individual events. Table II shows this for our above example. For the sake of illustration, we employ two alternative aggregation rules. The first uses the *average* number of participants across all news reports (1,440) and the *most frequent value* for security forces engagement (present, 1).[5] However, users preferring other aggregations can do so easily, as the second line shows. Here, we use the *maximum* reported number of protesters (3,000), and the *maximum* level of security forces engagement (physical intervention, 2). Of course, other aggregation rules are possible and can easily be applied by the user. One could compute confidence intervals around the aggregated numbers. In addition, one can rely on other variables in the aggregation process. For example, using date and time of when a report was released (not shown in the table), one could give preference to more recent reports.

Conclusion

Although most event data projects rely on media reports, few outline in detail how information is extracted from news reports and how this information is aggregated to individual events. This can negatively affect reliability and replicability of a coding project. We propose to introduce an additional coding step – the event report – that stores the extracted information before it is aggregated to the event level. Using an example from an event database on protest, we illustrate this procedure. The example shows that with slight changes to the conventional coding procedure and an automatic aggregation process, we can improve the coding process significantly. Event records increase transparency of the coding, allow users to go back to the original information and possibly re-interpret it, or even let them apply their own aggregation rules. This of course applies primarily to projects relying on more than one source. We hope that despite the increase in complexity this process entails, we have been able to convince readers of its advantages. The future release of our protest database will provide a live example of how this process is implemented.

Funding

We gratefully acknowledge funding for this project from the Alexander von Humboldt Foundation (Sofja Kovalevskaja Award) and from the EU FP7 Marie Curie Zukunftskolleg Incoming Fellowship Program, University of Konstanz (grant no. 291784).

Reference

Rød, Espen Geelmuyden & Nils B Weidmann (2013) Protesting dictatorship: The Mass Mobilization in Autocracies Database. Working paper, Department of Politics and Public Administration, University of Konstanz (http://www.cnc.uni-konstanz.de/research/mmad/).

Salehyan, Idean (2015) Best practices in the collection of conflict data. *Journal of Peace Research* 52(1): 105–109.

NILS B WEIDMANN, b. 1976, PhD in Political Science (ETH Zurich, 2009); Professor of Political Science, University of Konstanz (2012–).

ESPEN GEELMUYDEN RØD, b. 1985, MA (University of Oslo, 2012); PhD Fellow in Political Science, University of Konstanz (2012–); Research Assistant, Peace Research Institute Oslo (2012–); current main interest: dynamics of authoritarian regimes.

[5] The reported average number of participants omits the verbally specified numbers (hundreds and several thousand). In order to include them in the aggregation, a project could establish conventions for how to translate them into numbers.