Social Science Computer Review
1-11
© The Author(s) 2016
Reprints and permission:
sagepub.com/journalsPermissions.nav
DOI: 10.1177/0894439315625475
ssc.sagepub.com
SAGE

Supervised Event Coding From Text Written in Spanish: Introducing Eventus ID

Javier Osorio[1] and Alejandro Reyes[2]

Abstract

Recent innovations in conflict and computer research favor generating massive event data using automated coding protocols. Unfortunately, these approaches almost exclusively rely on English-language sources, thus causing problems of coverage bias and misleading inferences. In an effort to attenuate Anglocentrism in event data, we introduce Eventus ID, new software for supervised event coding from text written in Spanish. Drawing on real news reports, the application generates daily georeferenced data on how the military fights organized criminals in Mexico. Performance metrics show that Eventus ID is almost as accurate as humans for coding event data.

Keywords

event data, machine coding, Spanish language, Mexico

Introduction

Recent advancements at the intersection of conflict research and computer science enable innovative approaches for generating massive databases of event data and expanding our understanding about conflict processes (Bond, Bond, Oh, Jenkins, & Taylor, 2003; Leetaru & Schrodt, 2013; Schrodt, 2009, 2012b; Schrodt, Beieler, & Idris, 2014; Schrodt, Davis, & Weddle, 1994; Subrahmanian, 2013). Despite their global scope, most event coding projects primarily rely on information gathered from English-language sources. However, ignoring reports written in the native language of foreign locations is likely to induce coverage bias, reduce the quality of information, delay timely news, and ultimately degrade the accuracy of the event data. This is particularly problematic if we consider that 91.8% of the world population is non-English speaker (Lewis, Simons, & Fennig, 2013). Consequently, inferences derived from biased, incomplete, or inaccurately recorded events may generate misleading conclusions.

In an effort to offer alternatives to English-language, computer-generated data, this article introduces Eventus ID, an event coding software capable of identifying on who did what to whom, when and where from news reports from unstructured text written in Spanish.[1] Eventus ID's direct

[1] Department of Political Science, John Jay College of Criminal Justice, City University of New York, New York, NY, USA
[2] Instituto Nacional de Astrofísica Óptica y Electrónica, Tonanzintla, Puebla, Mexico

Corresponding Author:
Javier Osorio, John Jay College of Criminal Justice, City University of New York, New York, NY, USA.
Email: josorio@jjay.cuny.edu

Table 1. Verb Tenses in English and Spanish.

	Indicative		Subjunctive			
Person	Present	Past	Present	Imperfect	Gerund	Past Passive Voice
English						
I	arrest	arrested	arrest	arrested	arresting	was arrested
you	arrest	arrested	arrest	arrested	arresting	were arrested
he, she	arrests	arrested	arrests	arrested	arresting	were arrested
we	arrest	arrested	arrest	arrested	arresting	were arrested
you	arrest	arrested	arrest	arrested	arresting	were arrested
they	arrest	arrested	arrest	arrested	arresting	were arrested
Spanish						
yo	*arresto*	*arresté*	*arreste*	*arrestara o arrestase*	*arrestando*	*fui arrestado*
tú	*arrestas*	*arrestaste*	*arrestes*	*arrestaras o arrestases*	*arrestando*	*fuiste arrestado*
ella, él, usted	*arresta*	*arrestó*	*arreste*	*arrestara o arrestase*	*arrestando*	*fue arrestada o fue arrestado*
nosotros	*arrestamos*	*arrestamos*	*arrestemos*	*arrestáramos o arrestásemos*	*arrestando*	*fuimos arrestados*
vosotros	*arrestáis*	*arrestasteis*	*arrestéis*	*arrestarais o arrestaseis*	*arrestando*	*fuisteis arrestados*
ellas, ellos, ustedes	*arrestan*	*arrestaron*	*arresten*	*arrestaran o arrestasen*	*arrestando*	*fueron arrestadas o fueron arrestados*
vos	*arrestás*	*arrestaste*	*arrestés*	*arrestaras o arrestases*	*arrestando*	*fuiste arrestado*

predecessor is Tabari, the most popular software for event coding from reports written in English (Schrodt, 2009). Eventus ID shares some coding conventions with its forebear, yet it offers two key innovations: the use of two simultaneous event coding algorithms for processing complex grammatical structures and the incorporation of an event locator protocol for georeferencing the data at the subnational level. To the best of our knowledge, this is the first program for coding events from Spanish texts. This software aims to reduce the often overwhelming time and financial requirements for manually coding event data (Baumgartner, Jones, & MacLeod, 1998) and seeks to facilitate the creation of customized databases in comparative politics, international relations, and sociology relying on text written in Spanish.

Coding Event Data From Text Written in Spanish

Simply stated, an event provides information about someone (*source*) doing something (*action*) to someone else (*target*) in a certain place and time. To detect the source and target, Eventus ID uses dictionaries of actors developed by the researcher. Additionally, a dictionary of verbs identifies actions. While reading the text, Eventus ID uses the actors and verbs dictionaries as searching categories to identify relevant information. Eventus ID also identifies dates and relies on a dictionary of locations to detect places. Once these elements are detected, the program stores the textual information in numeric format.

Coding events in Spanish presents two main difficulties. The first is the complexity of conjugating verb tenses. English grammar allows simple verb conjugation by adding "s," "ed," or "ing" to the infinitive. In addition, the gender and number generally do not affect the verb form. Consequently, the person (e.g., you, she, and we) can be indistinctly combined with different verb forms. These simple grammatical rules allow Tabari's stemming algorithm to automatically identify all the different verb tenses. For example, Panel (a) in Table 1 shows that "arrest" is easily conjugated for different tenses, gender, and numbers by simply adding a suffix to the stem. In this way, Tabari identifies a variety of verb forms with minimal computational effort.

Although convenient for coding in English, Tabari's stemming facility is not appropriate for coding in Spanish because verb conjugation in the latter integrates the gender and number of the subject in the tense, thus making it more complex. As shown in Panel (b) of Table 1, the ending part of the verb "*arrestar*" (to arrest) varies substantially across tenses, number, and gender. The "shortcuts" that might be useful for coding in English would be counterproductive in Spanish as they would substantially increase coding error and the computational demands for conjugating all tenses. Given the complexity of Spanish, Eventus ID does not include a stemming algorithm. Unfortunately, this strategy for reducing coding error requires users to devote more efforts to developing detailed verb dictionaries.

The second challenge refers to the frequent use of the present indicative in journalistic writing style. In Spanish, the present indicative is formed by removing the infinitive ending of the verb (e.g., taking out the final "*ar*" from the verb *arrestar*) and replacing it with an ending that indicates the person performing the action. In this way, the conjugation already gives information about the person as part of the verb, and in consequence the subject is omitted. To make things even more complex, the present indicative is often used for referring to events that occurred in the past. Thus, while the sentence "*arrestan a un criminal*" literally refers to an action carried out in the present, it also refers figuratively to a past event. By omitting the subject from the sentence, the present indicative makes event coding more difficult. The reason is that the sentence would not contain a source, thus breaking the fully specified source–action–target event structure. Due to the pervasiveness of the present indicative in journalistic narratives in Spanish media (Guízar García, 2004; Martínez, Miguel, & Vázquez, 2004), coding events requires a protocol capable of adapting to the complexities of this language.

Eventus ID Coding Process

In contrast to fully automated methods[2] that do not require human involvement, Eventus ID's supervised method requires the researcher's intervention in different stages, particularly in dictionary development and validation. Figure 1 illustrates the six steps of the coding process.

1. Information gathering. In Step 1.1, users can us Rich Site Summary readers or web scrappers to automatically extract content form the web. Alternatively, researchers can rely on humans to select and download specific documents. The latter approach improves the validity and accuracy of the outcome. However, it implies a trade-off between the speed and nondiscrimination of automatic methods and the precision, yet slow pace, of manual selection.
2. Corpus of text. Eventus ID comes with ancillary software for compiling and formatting web-extracted documents into a single file, named corpus (Step 2.1). The format breaks each document into its various paragraphs and assigns a label per document paragraph. The corpus constitutes the input for the coding process (Step 2.2).
3. Event identification. The processes of event coding (Stage 3) and data georeferencing (Stage 4) are integrated into Eventus ID, yet these two stages are discussed separately for illustration purposes. The software requires researchers to develop dictionaries of actors and verbs (Steps 3.1 and 3.2) to identify the source, action, and target in the corpus. Building specialized dictionaries is the key for customizing projects to specific topics while considering language regionalisms across Spanish speaking countries. The protocol uses two event coding algorithms that better accommodate to Spanish (Step 3.3) and generates the output (Step 4.3). Named Entity Recognition (NER) software (The Stanford Natural Language Processing

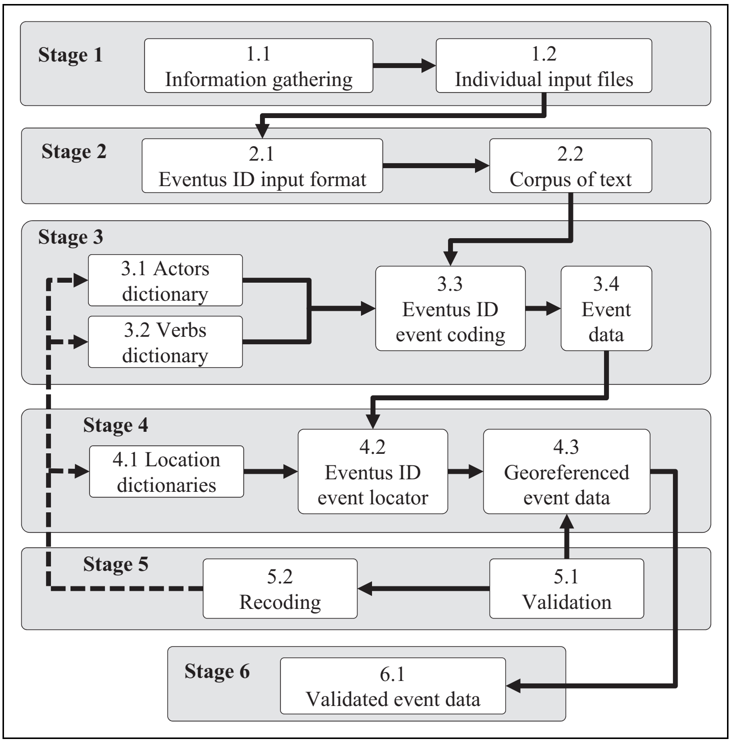
Stage 1
1.1 Information gathering
1.2 Individual input files
Stage 2
2.1 Eventus ID input format
2.2 Corpus of text
Stage 3
3.1 Actors dictionary
3.2 Verbs dictionary
3.3 Eventus ID event coding
3.4 Event data
Stage 4
4.1 Location dictionaries
4.2 Eventus ID event locator
4.3 Georeferenced event data
Stage 5
5.2 Recoding
5.1 Validation
Stage 6
6.1 Validated event data

Figure 1. Eventus ID coding process.

Group, 2014) can assist in developing actor dictionaries by automatically identifying names when they are unknown to the researcher. However, NER is not designed for detecting verb sentences.

4. Event location. This stage requires the development of location dictionaries containing the names of primary (states) and secondary (municipalities) localities to be used as search categories, along with some filters to enable geographic disambiguation (Step 4.1). The georeferencing algorithm uses locality names to inspect the corpus and identify the places where the events occurred (Step 4.2). The output is a georeferenced database of event data (Step 4.3).
5. Validation. The conventional approach for assessing accuracy of the outcome is to compare a sample of manually coded data against the computer-generated events. Discrepancies between manual and computer outcomes provide insights for modifying the actor, verb, or location dictionaries. Validation is often labor intensive, and a series of coding–validation–recoding iterations (Steps 5.1 and 5.2) are often necessary to improve the accuracy of the data.
6. Output. The final output is a validated database of georeferenced event data (Step 6.1).

Coding Innovations

Event Coding Algorithms

In contrast to Tabari that works with a single event coding algorithm, Eventus ID uses two pattern recognition algorithms: the general sequence algorithm (GSA) codes events that comply with the

Table 2. Event Coding Algorithms.

General Sequence Algorithm	Partial Sequence Algorithm
Step (1) Search for the actor -Load the actor dictionary -Search for the longest actor first -When an actor is found, store as `actor1` and pause the search Step (2) Search for the verb - Load the verb dictionary - Resume reading from previous pause - Search for the longest verb first - When a verb is found, store as `verb` and pause the search - If no verb is found, go to Step 4 Step (3) Search for the actor - Reload the actor dictionary - Resume reading from previous pause - Search for the longest actor first - When an actor is found, store as `actor2` and pause the search Step (4) Save the event - Save [`actor1`] [`verb`] [`actor2`] in database - If no verb is found, save the event as [`actor1`] [◊][◊] in the database - Start again from Step 1	Step (1) Search for the verb - Load the verb dictionary - Search for the longest verb first - When a verb is found, store as `verb` and pause the search Step (2) Search for the actor - Load the actor dictionary - Resume reading from previous pause - Search for the longest actor first - When an actor is found, store as `actor2` and pause the search - If no actor is found, go to Step 3 Step (3) Save the event - Save [◊] [`verb`] [`actor2`] in the database - If no actor is found, save the event as [◊] [`verb`] [◊] in the database - Start again from Step 1

Note. [◊] represents a blank space.

full source–action–target structure and the partial sequence algorithm (PSA) codes incomplete events that omit the source and present a verb–target structure. PSA is particularly helpful for processing sentences using the present indicative verb form. Both coding algorithms begin by identifying the date of the event (`date`), followed by the document name and specific paragraph in the text corpus (`FileName_P1_P2`). As indicated in Table 2, each algorithm then uses its own recognition sequence to identify events in the corpus. GSA starts coding an event by searching for the source (coded as `actor1`), then the action (coded as `verb`), and finally the target (coded as `actor2`). PSA skips the source and starts searching for the action (`verb`) and then looks for the target (`actor2`). Both searching sequences are implemented simultaneously. Eventus ID saves the event-coding outcome separating its elements by tabs (denoted by →). The algorithms respectively generate the following outcomes:

GSA: `date` → `FileName_P1_P2` → `actor1` → `verb` → `actor2`
PSA[3]: `date` → `FileName_P1_P2` → → `verb` → `actor2`

To illustrate how GSA works, consider the following sentence: [4]

> Army troops arrested a member of a criminal group.
> *Tropas del ejercito arrestaron a miembro de un grupo criminal.*

In this example, all three elements of the event (source–action–target) are part of the sentence in the required order. In consequence, GSA identifies “Army troops” as the source, “arrested” as the action, and “member of a criminal group” as the target.

GSA can also identify sentences in passive voice, a common verb tense in Spanish. Instead of having a regular subject–verb–object structure, passive voice inverts the order of the subject and object and yields to an object–verb–subject structure as illustrated below:

A member of a criminal group was arrested by Army troops.

Miembro de un grupo criminal fue arrestado por tropas del Ejercito.

GSA does not requires the three elements of source–action–target to be present in the sentence for Eventus ID to register the event. This feature is useful when the reports include lists of nouns. For example:

The Army seized an arsenal containing AK-47 rifles, R-15 assault rifles, grenade launchers, ammunition, and communication devices.

El Ejercito decomiso un arsenal que contenia rifles AK-47 y R-15, lanza granadas, municiones y equipo de comunicacion.

Eventus ID would code the source ("Army"), the action ("seized"), and the target ("arsenal") as one event and then a list of nouns ("AK-47 rifles," "R-15," "grenade launchers," "ammunition," and "communication devices") in separate event lines.

The following sentence illustrates a grammatical structure that fits the PSA coding scheme. Since the translation of present indicative from Spanish into English obscures the nuances of the present indicative verb form, the next example is only presented in Spanish: *Arrestan a un criminal.*

The subject is omitted from the sentence because of the conjugation of the verb in present indicative tense. In the absence of the subject, Eventus ID uses PSA to identify "*arrestan*" (to arrest) as the action, and "*a un criminal*" (a criminal) as the target. Researchers could then decide whether to fill the omitted source with ad hoc codes in the postcoding stage.[5] In this way, GSA and PSA coding schemes of Eventus ID adapt to the grammatical complexities of Spanish.

Geolocation Algorithm

In contrast to Tabari that lacks an integrated geolocation feature, Eventus ID's event location algorithm (ELA) uses dictionaries of states and municipalities to identify the geographic occurrence of events at two subnational levels, thus generating highly disaggregated spatial data. To develop location dictionaries, researchers can use lists of toponyms provided by government agencies or use NER geographic classifiers (The Stanford Natural Language Processing Group, 2014). Finally, due to the complexities of georeferencing computerized event data (Chojnacki, Ickler, Spies, & Wiesel, 2012; Hammond & Weidmann, 2014), ELA includes a filter dictionary to prevent certain words from being erroneously classified as locations.

Table 3 outlines the event location procedure. ELA uses the event data generated in Stage 3 (see Figure 1) to recall the specific paragraph from which each event was extracted. It then reads the entire corpus to locate that precise paragraph. Next, ELA uses the location dictionaries to detect the name of a state or municipality in the paragraph. If a location is identified, the filters verify whether the location should be kept or discarded. If the location is not filtered, the program saves the locality in the database. If the paragraph contains no location, ELA expands the search to the rest of the news report starting from the first paragraph of the document. If a location is recognized in the document, the protocol checks whether it should be filtered or not. If it passes the filter, ELA saves the location code in the database. If no location is detected in the document, the protocol stops the search and moves to the next event in the database.

To illustrate how the ELA works, consider the following paragraph:

Troops deployed in the municipality of San Luis Rio Colorado, Sonora seized 227 packages of marijuana.

Tropas destacamentadas en el municipio de San Luis Rio Colorado, Sonora decomisaron paquetes de marijuana.

Table 3. Geolocation Algorithm.

```
(1) Identify an event
        - Load the event database.
        - Select an event from a new line.
        - Identify the paragraph name (FileName_P1_P2) and use it as searching criteria.
(2) Identify the paragraph in the text corpus.
        - Load the text corpus.
        - Search for the paragraph from which the event was extracted.
(3) Search for the location of the event in the paragraph.
        - Load the location dictionaries (states and municipalities).
        - Use the items of the location dictionaries as searching criteria.
        - Start searching for the location in the source paragraph.
        - If the location is found, store the code.
        - Keep searching for locations and storing them until the end of the paragraph.
        - If there are no more locations in the paragraph, then go to Step 5.
        - If no location is found in the paragraph, go to Step 4.
(4) Expand the search to the rest of the document.
        - Select the remaining paragraphs belonging to the same document (FileName).
        - Search for the location in all paragraphs of the document.
        - Begin searching in the first paragraph.
        - If the location is found in the document, store the location code and go to Step 5.
        - If the location is not found in the document, stop searching and go to Step 1.
(5) Filter the location.
        - Load the filters dictionary.
        - Verify that the location identified does not match any item in the filters dictionary.
        - If the location matches a filter, go back to Step 3.
        - If the location does not match a filter, go to Step 6.
(6) Save the location.
        - Save the location at the end of the coded event line in the event database.
        - Start again from Step 1.
```

Given the right set of actors, verbs, and location dictionaries, Eventus ID recognizes the key components of the event as "troops" (source), "seized" (action), and "marijuana" (target). In addition, the location algorithm identifies the municipality "San Luis Rio Colorado" in the state of "Sonora" as the location where the event took place. In this way, Eventus ID produces a full event data account indicating who, did what, to whom, when, and where.

Application

Eventus ID is a generic supervised event coding protocol developed for assisting researchers in generating customized event databases from Spanish texts. To illustrate the software, this section presents an application to generate event data of counter-narcotic efforts conducted by the Mexican Army (*Secretarıá de la Defensa Nacional*, SEDENA) using real military press releases issued between December 2012 and January 2013. The decision to use these documents is based on the possibility of replicating this example without inflicting copyrights. As Schrodt (2014) warns, a central challenge for the transparency and validity of automated coding pertains to the intellectual property rights that prohibit researchers from sharing copyrighted materials. The lack of original reports inhibits the possibility of testing computerized coding protocols using real corpuses and often forces scholars to use a fake corpus or not to release any corpus at all. To overcome this challenge, this application has the permission of SEDENA to use real press releases as corpus of text.[6]

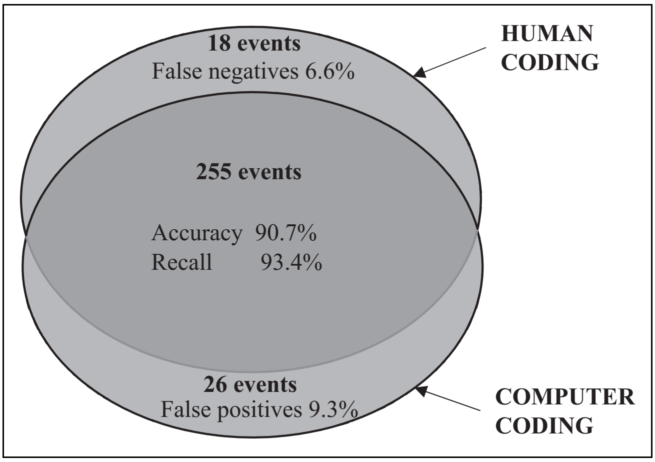

Figure 2. Validity assessment of event coding.

Detailed and encompassing dictionaries are crucial for generating high-quality event data. In this sense, supervised event coding is not a panacea. It requires researchers to devote substantial effort for developing customized dictionaries (Best, Carpino, & Crescenzi, 2013).[7] In this particular application, the actor dictionary consists of 140 nouns grouped in 6 categories: military personnel, unidentified individuals, criminals, drugs, weapons, criminal assets, and others. The verb dictionary contains a list of 26 verbs organized in 4 action types: arrests, seizures, destruction, and rescue. The location dictionaries comprise the names of all states and municipalities (Instituto Nacional de Estadística y Geografía, 2011) and a list of filters. Eventus ID uses these different dictionaries to identify events conducted by the Mexican Army against criminal organizations.

Coding accuracy is a central concern in computerized textual annotation (Grimmer & Stewart, 2013; King & Lowe, 2003; Schrodt & Gerner, 1994). To validate the precision of the computer output, a team of human coders followed Eventus ID's coding sequence to process the entire corpus and identified a total of 273 georeferenced events. Human coding thus serves as the "gold standard" to cross-check the correctness of machine data, a regular practice in validating textual annotation (Cardie & Wilkerson, 2008). After iterative processes of event coding, assessment, elimination of duplicates, disambiguation, and recoding, [8] the final computer outcome identified a total of 281 georeferenced events at the daily municipal level.

Following the standard performance evaluation metrics (Cardie & Wilkerson, 2008), Figure 2 shows the extent of congruence between the manual and computerized event data. The criteria for correctness use an astringent benchmark, in which an event is considered correctly coded if *all* the elements of the computer-generated event match the source, action, target, location (at both the state and municipal levels), and the date identified by human coders. Both protocols agree on 255 events. *Accuracy* refers to the percentage of events proposed by the system that are correctly identified in comparison to the manually coded database. The result shows that Eventus ID's accuracy is up to 90.7% when compared to human coders. *Recall* is the percentage of the manually classified events that are correctly identified by the system, which amounts to 93.4%.[9] In contrast, the automated procedure failed to identify 18 events that were detected in the manual coding process. The category *false negatives* refers to the share of manually coded events that the system missed, which is only 6.6%. Finally, the computerized outcome generated 26 events that were not previously identified by human coders. These *false positives* correspond to 9.3% of events proposed by the system that humans did not identify.

There is no consensus in the standards of computerized coding accuracy (Cardie & Wilkerson, 2008). At minimum, automated coding should perform better than random chance.[10] At maximum, computer-generated events should match human coders at 100%. The output presented in this article is close to the upper end of accuracy, thus suggesting that Eventus ID is almost as precise as humans for coding events. This application also performs better than Eventus ID's beta version, which reported 82% accuracy (Osorio, 2013, 2015). It is also more precise than the 75% to 85% accuracy of databases using Tabari (Best et al., 2013; Schrodt, 2009; Schrodt & Gerner, 1994) and as accurate as The Reader (King & Lowe, 2003).[11] Besides its precision, Eventus ID coded the entire corpus of 102 kilobytes in only 25.16 sec, a miniscule fraction of the several hours that took humans to manually code the corpus.[12]

Discussion

The massive availability of global news digital media and the development of technology for automated event coding opened the possibility for understanding a broad range of conflict processes throughout the world. Accurately depicting the characteristics of specific events that conform to aggregated trends of conflict is of paramount relevance for both academic and policy communities. Scholars need precise data to derive valid inferences about political phenomena, as much as the policy sector needs reliable information for decision making. Unfortunately, most efforts to understand conflict in foreign countries using automated event coding techniques have the limitation of relying almost exclusively on English-written news reports (Weller & McCubbins, 2014). Neglecting valuable information produced in the native language of foreign locations likely degrades the quality of the metrics used to analyze conflict processes and, ultimately, affects the conclusions drawn from the data. Eventus ID contributes to ameliorating the problem of Anglocentrism in computer generated event data by allowing researchers the possibility to code events from Spanish texts.

Eventus ID is capable of generating an accurate categorization and geolocation of event data based on Spanish news reports. This example provides detailed events about different counternarcotic tactics conducted by the Mexican Army against drug trafficking organizations at the municipal-daily level. Moving beyond this initial application, the software offers researchers the possibility to generate their own event data on a broad variety of topics in Latin America using native information sources. This technological innovation thus broadens the opportunities to better understanding political phenomena in the region.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: National Science Foundation [SES-1123572]; the Jennings Randolph Peace Scholar fellowship of the United States Institute for Peace; the Drugs Security and Democracy fellowship of the Social Science Research Council – Open Society Foundations;the Program on Order, Conflict and Violence at Yale University; the Mario Einaudi Center for International Studies at Cornell University; and the Dissertation Fellowship of the Harry Frank Guggenheim Foundation.

Notes

[1] The replication files and supplementary materials are available at: https://github.com/javierosorio/SSCR_2016_replication

2. For a review of computerized textual annotation methods, see Cardie and Wilkerson (2008), Grimmer and Stewart (2013), and Schrodt and Gerner (2012).
3. Notice that due to the subject suppression in present indicative, there is no `actor1` between the two tabs → → in the PSA outcome.
4. Spanish text examples in this manuscript deliberately omit accents.
5. For example, researchers can impute "Authorities" as an ad-hoc source of this event, as the state is the only actor who can arrest a criminal.
6. Note to the reviewer: if accepted, the software documentation will provide the citations of the press releases used in the corpus.
7. Researchers might also rely on pre-existing dictionaries such as CAMEO (Gerner, Schrodt, Yilmaz, & Abu-Jabr, 2002; Schordt 2012a) or use NER software to identify possible actor names.
8. This process relates to Stage 5 in Figure 1, and feedback loops in Stages 3 and 4.
9. Consider H as the total number of events identified by humans, M as the total number of machine generated events, and C as the total number of events correctly identified by the computer when compared to human coding. The *accuracy* (A) metric is $A = \frac{C}{M}$ and *recall* (R) is $R = \frac{C}{H}$.
10. The probability of correctly coding all five characteristics of an event ($i =$ source, action, target, date, location) by random chance ($p = .5$) is $p^i = 3.12\%$.
11. Although different dictionaries prevent direct comparisons across distinct databases, the overall accuracy metrics allow comparing their coding performance.
12. This comparison does not include the time invested in dictionary development, a task necessary for both coding approaches.

References

Baumgartner, F., Jones, B., & MacLeod, M. (1998). Lessons from the trenches: Ensuring quality, reliability, and usability in the creation of a new data source. *The Political Methodologist*, *8*, 1–10.

Best, R. H., Carpino, C., & Crescenzi, M. J. C. (2013). An analysis of the TABARI coding system. *Conflict Management and Peace Science*, *30*, 335–348.

Bond, D., Bond, J., Oh, C., Jenkins, C. J., & Taylor, C. L. (2003). Integrated data for events analysis (IDEA): An event typology for automated events data development. *Journal of Peace Research*, *40*, 733–745.

Cardie, C., & Wilkerson, J. (2008). Text annotation for political science research. *Journal of Information Technology and Politics*, *5*, 1–6.

Chojnacki, S., Ickler, C., Spies, M., & Wiesel, J. (2012). Event data on armed conflict and security: New perspectives, old challenges, and some solutions. *International Interactions*, *38*, 382–401.

Gerner, D., Schrodt, P. A., Yilmaz, O., & Abu-Jabr, R. (2002). The creation of CAMEO (Conflict and Mediation Event Observations): An event data framework for a post cold war world. In *Annual Meeting of the American Political Science Association*. Retrieved January 10, 2013, from http://web.ku.edu/keds/papers.dir/Gerner.APSA.02.pdf

Grimmer, J., & Stewart, B. M. (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, *21*, 267–297.

Guízar García, E. (2004). El uso de los verbos en los titulares de cinco diarios de la ciudad de México: análisis sintáctico. Doctoral dissertation, Universidad Nacional Autónoma de México, Mexico city, Mexico.

Hammond, J., & Weidmann, N. B. (2014). Using machine-coded event data for the micro-level study of political violence. *Research and Politics*, *1*, 1–8.

Instituto Nacional de Estadística y Geografía. (2011). Marco geoestadístico nacional. Retrieved from http://www.inegi.org.mx/geo/contenidos/geoestadistica/default.aspx

King, G., & Lowe, W. (2003). An automated information extraction tool for international conflict data with performance as good as human coders: A rare events evaluation design. *International Organization*, *57*, 617–642.

Leetaru, K., & Schrodt, P. A. (2013). GDELT: Global data on events, location and tone, 1979-2012. Retrieved from http://data.gdeltproject.org/documentation/ISA.2013.GDELT.pdf

Lewis, M. P., Simons, G. F., & Fennig, C. D. (2013). Summary by language size. Retrieved from http://www.ethnologue.com/statistics/size

Martínez, F., Miguel, L., & Vázquez, C. (2004). La titulación en la prensa gráfica. Retrieved from http://www.perio.unlp.edu.ar/grafica1/htmls/apuntescatedra/apunte_titulacion.pdf

Osorio, J. (2013). Hobbes on drugs: Understanding drug violence in Mexico. Doctoral Dissertation, University of Notre Dame. Retrieved from http://www.javierosorio.net//#!research/c240r

Osorio, J. (2015). The contagion of drug violence: Spatio-temporal dynamics of the Mexican war on drugs. *Journal of Conflict Resolution*, *59*, 1403–1432.

Schrodt, P. A. (2009). TABARI. Textual analysis by augmented replacement instructions. Retrieved from http://eventdata.parusanalytics.com/software.dir/tabari.html

Schrodt, P. A. (2012a). CAMEO: Conflict and mediation event observations. Event and actor codebook. Retrieved from http://eventdata.parusanalytics.com/cameo.dir/CAMEO.Manual.1.1b3.pdf.

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

Schrodt, P. A. (2014). The legal status of event data. Retrieved from http://asecondmouse.wordpress.com/2014/02/14/the-legal-status-of-event-data/

Schrodt, P. A., Beieler, J., & Idris, M. (2014). *Three's a charm?: Open event data coding with EL: DIABLO, PETRARCH, and the open event data alliance*. Toronto, Canada: International Studies Association. Retrieved from http://parusanalytics.com/eventdata/papers.dir/Schrodt-Beieler-Idris-ISA14.pdf

Schrodt, P. A., Davis, S. G., & Weddle, J. L. (1994). Political science: KEDS-A program for the machine coding of event data. *Social Science Computer Review*, *12*, 561–587.

Schrodt, P. A., & Gerner, D. (1994). Validity assessment of a machine-coded event data set for the Middle East, 1982-1992. *American Journal of Political Science*, *38*, 825–854.

Schrodt, P. A., & Gerner, D. (2012). Fundamentals of machine coding. In *Analyzing international event data: A handbook of computer-based techniques*. Retrieved from http://parusanalytics.com/eventdata/papers.dir/AIED.Preface.pdf

Subrahmanian, V. S. (2013). *Handbook of computational approaches to counterterrorism*. New York, NY: Springer.

The Stanford Natural Language Processing Group. (2014). Stanford named entity recognizer. Retrieved from http://nlp.stanford.edu/software/CRF-NER.shtml

Weller, N., & Kenneth, M. (2014). *Raining on the parade: Some cautions regarding the global database of events, language and tone dataset*. Retrieved January 10, 2013 from http://politicalviolenceataglance.org/2014/02/20/raining-on-the-parade-some-cautions-regarding-the-global-database-of-events-language-and-tone-dataset/

Author Biographies

Javier Osorio is an assistant professor in the Department of Political Science at John Jay College of Criminal Justice, CUNY. He received his Ph.D. in Political Science from the University of Notre Dame. His research agenda is located at the intersection of political violence, quantitative methods and computer science.

Alejandro Reyes is a software developer and currently works as System Specialist for Audi. He received his MS in Computer Science from Instituto Nacional de Astrofísica Óptica y Electrónica, Mexico. His specializes is in natural language processing, voice recognition, systems for question answering and robotics.