Creating Custom Event Data: A Bag-of-Tricks

Andrew Halterman[*], Andreas Beger[†], Benjamin E. Bagozzi[‡],
Philip A. Schrodt[§], Grace I. Scarborough[¶]

October 2, 2026

Word Count: 5,905 (excluding references, supplemental material, and tables)

Abstract

Political scientists who employ event data—structured records of "who did what to whom"—would ideally use custom datasets tailored to their questions, but the cost of producing custom data pushes them toward existing general event datasets. We introduce a modular framework for producing custom event data that decomposes the process into five independently customizable steps: event detection, event "attribute" extraction (actors, recipients, dates, and locations), entity linking, entity categorization, and date/location resolution. Researchers can use models we provide and invest effort only in the steps their project requires. To address the two most challenging steps, we develop new models: a small language model for extracting event attributes, trained on synthetic and real data, that generalizes to new event types; and a Wikipedia-based entity linking system that disambiguates and categorizes political entities. We validate the pipeline against held-out expert and human-adjudicated annotations and by evaluating it, without retraining, against the hand-coded Electoral Contention and Violence dataset. Applied to ECAV, the pipeline matches ECAV's actor category for 62% of events and 45% for recipients. Given ECAV's own spans, categorization reaches 89%, suggesting that customization effort should focus on extraction. The pipeline is open source, runs on consumer hardware, and requires no proprietary APIs.

[*]Michigan State University, ahalterman0@gmail.com
[†]Independent Researcher, andy@basilanalytics.com
[‡]University of Delaware, bagozzib@udel.edu
[§]Parus Analytics LLC, schrodt735@gmail.com
[¶]Leidos, grace.i.scarborough@leidos.com

1 Introduction

Political scientists often draw on event data derived from text to answer questions about the actions and interactions of political entities. Event data consists of structured data on event types, the actors and recipients involved in the event, and the dates and locations of events, or, more colloquially, is structured data that answers the questions "who did what to whom, where and when?" Researchers can draw on large, off-the-shelf event datasets, both hand-coded—such as ACLED (Raleigh, Kishi and Linke, 2023), MID (Palmer et al., 2022), or UCDP GED (Sundberg and Melander, 2013)—and machine-coded—such as GDELT (Leetaru and Schrodt, 2013), POLECAT (Scarborough et al., 2023), the historical Phoenix event data (Althaus et al., 2020), or ICEWS (Boschee et al., 2015). While using off-the-shelf datasets is low cost and makes many projects feasible, this approach has its drawbacks. Pre-built datasets might be several years out of date or may not be updated anymore at all, limiting the range of historical data that they cover or making it difficult to use such data in forecasting applications. Off-the-shelf datasets may not cover the news sources or geographic areas that a researcher wishes to study. Or, most importantly, these data resources may not code the events or actors that a researcher requires. If the concepts researchers intend to measure are not well aligned with existing event data options, using such data can raise significant measurement validity concerns (Adcock and Collier, 2001; Goertz, 2020).

Thus, for many political science research projects it is preferable to construct a custom event dataset specifically tailored to a specific question or application (e.g., Toft and Zhukov, 2015; Zhukov and Ayers, 2023; Salau and Halterman, 2025). However, creating custom event datasets is often prohibitively costly. Hand-coding event data from text is an expensive, multi-year undertaking. Automated techniques for producing event data are difficult or expensive to use, require technical expertise and effort to customize them, and are often unacceptably inaccurate. This paper addresses these challenges.

This paper makes two contributions. First, we propose a modular framework that decomposes event data production into five independently customizable steps: event detection, attribute extraction (identifying the actors, recipients, dates, and locations of events), entity linking, entity categorization, and date/location resolution. The framework clarifies the design decisions researchers face and allows them to adopt pretrained models where applicable, investing effort only in the steps their project requires. Second, we develop new models for the two most challenging steps: a small language model for extracting event attributes from English language text, trained on a mixture of synthetic and real text so that it generalizes to new event types without hand annotation; and a Wikipedia-based entity linking system that disambiguates

political actors without hand-built dictionaries.[1] The five-part framework is general to any event ontology, while the trained components we release were built around the common PLOVER event ontology. We demonstrate that these models can generalize to new event types, including by validating the pipeline, with no retraining, against the hand-coded Electoral Contention and Violence (ECAV) dataset (Daxecker, Amicarelli and Jung, 2019), which uses an entirely different ontology.

We provide our tools as an open-source Python package, the Next Generation Event Coder (NGEC), for producing event data from English-language text.[2] The pipeline uses only open-weight models, can run entirely offline on consumer hardware, and supports customization at multiple levels: researchers can use our pretrained models with new text sources, define new event types by training document classifiers, or extend the system with new attributes. We provide guidance on each customization scenario and estimate the effort required. As we emphasize throughout the paper, these steps can each be used on their own to solve specific research problems.

2 Existing Approaches to Automated Event Extraction

The process of producing an event record from text consists of taking in a document, identifying information from within the document about the events and entities reported in the document, and structuring that information according to an ontology. Consider the simplified example in Figure 1. Given an input document, an initial phase of the event extraction system might identify an event description, then identify the spans of text[3] that report the entities involved in the event. Finally, to be useful in downstream analysis, the parts of an event are categorized according to the ontology being used.[4] This includes categorizing actors and recipients, and resolving locations and dates to canonical values.

Creating custom event data consists of answering several project-specific questions before beginning the coding process: what event types should one code and how should they be defined? (e.g., should "riots" be distinguished from "protests"?) What attributes do events have? (e.g., should a protest event have distinct "participants" and "organizer" attributes?). How should entities

[1] An earlier version of the pipeline we describe here generated the global POLECAT dataset (Halterman et al., 2023; Scarborough et al., 2023).

[2] https://github.com/ahalterman/NGEC-2025

[3] A span is a verbatim sequence of words from the document, such as "Syrian government forces" or "Tuesday."

[4] While our pipeline is agnostic to the specific ontology being used, we will use the fairly generic PLOVER ontology (Halterman et al., 2023) for most examples in the paper.

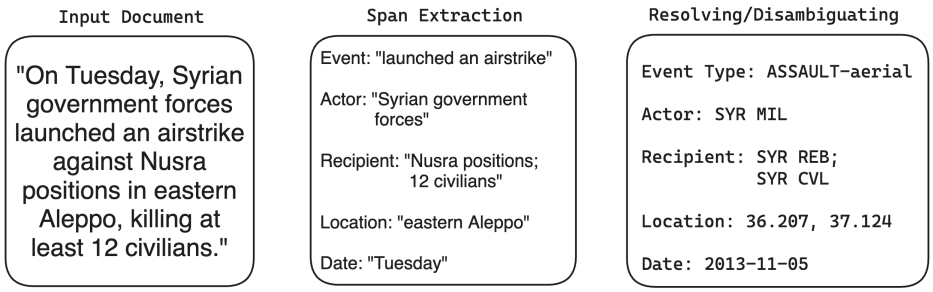


Figure 1: *A high level overview of event data extraction, emphasizing the distinction between information extraction steps and resolution/disambiguation. For example, a coding pipeline must identify that "Nusra" is the target of the event, and separately that Nusra is an instance of a SYR[ian] REB[el] group (according to the researcher's ontology), potentially using information from an external data source such as Wikipedia.*

be aggregated? (e.g., should all law enforcement entities be resolved to a single "police" category, or should local and national police be distinguished in the dataset)? We return to this in Section 4.

Many existing projects and methods in social science and computer science have sought to build automated systems to extract event information from text (see, especially Zhukov and Ayers, 2023; Haunss et al., 2025). However, none of these systems have enabled applied social scientists to feasibly generate *custom* event data.

Dictionary-based methods have long been a technique for producing event data in social sciences (Schrodt, Davis and Weddle, 1994; Gerner et al., 2002; Schrodt, 2009; O'Brien, 2010; Norris, Schrodt and Beieler, 2017). These systems use pattern matching against hand-built lists of terms to perform event identification, actor identification, and actor resolution simultaneously. While researchers have built custom dictionary-based coders for specific event types (Brathwaite and Park, 2018) or non-English languages (Osorio and Reyes, 2017; Liang et al., 2018), and developed techniques for semi-automated dictionary development (Halterman et al., 2018; Radford, 2019), dictionary-based coders face limitations. They suffer from low recall from incomplete dictionaries and their reliance on exact matching, context unawareness (e.g., "launch a strike" could refer to military or labor actions), and the need to maintain exhaustive, time-specific actor dictionaries. These difficulties have limited the adoption of dictionary-based methods by researchers seeking to produce custom event datasets.

A second body of work in computational linguistics treats event extraction as identifying *linguistic frames or arguments* (Baker, Fillmore and Lowe, 1998; Das et al., 2014; Doddington et al., 2004; Palmer, Gildea and Kingsbury, 2005). More recently, neural architectures and LLMs have achieved strong per-

formance on extraction benchmarks through document-level models (Li, Ji and Han, 2021; Ebner et al., 2020; Alsarra et al., 2025), question-answering formulations (Du and Cardie, 2020), and generative approaches (Hsu et al., 2022; Dai, Radford and Halterman, 2022). However, these systems focus specifically on argument extraction—identifying text spans corresponding to event-specific arguments—without providing the broader steps political scientists need, including linking extracted names to canonical identifiers, categorizing actors into substantively meaningful groups, and resolving locations to coordinates. A social scientist cannot take a trained extraction model and produce a usable event dataset without building the remaining infrastructure themselves.

Third, researchers are increasingly adopting *large language models* to identify events in text (Zhukov and Ayers, 2023; Halterman and Keith, 2025), including through broadly useful fine-tuned language models (Hu et al., 2022; Meher and Brandt, 2025). Treating event detection as document classification is attractive because it is well understood in social science, straightforward to customize, and amenable to synthetic data augmentation (Halterman, 2025).

While prior work on automated event extraction has made important contributions to coding event data, it has left three gaps for applied researchers:

- No clear modular framework that decomposes event extraction into independently customizable steps, allowing researchers to invest effort only where their project requires it.
- No efficient, general-purpose method for extracting the actors, recipients, dates, and locations of events that generalizes to new event types without retraining.
- No technique for disambiguating political entities and categorizing them into researcher-defined groups without maintaining hand-built dictionaries.

The remainder of the paper addresses these gaps.

3 A Five-Part Framework for Coding Event Data

Our first contribution is a proposed five-step conceptual framework for coding events, independent from the specific ontology being used.[5] Posing event extraction in this way shows applied researchers where they can use existing models or where they need to customize specific steps to produce their custom event data. It also highlights that certain steps can be used on

[5] Ahn (2006) suggests an alternative four-part conceptual framework, but focusing on computational linguistics challenges in the ACE dataset, rather than on practical guidance for social scientists creating custom event data. Haunss et al. (2025) also argue for the importance of modular pipelines in coding protest event data.

their own to address other research requirements, such as resolving named entities to their Wikipedia pages.

1. **Event detection**: We begin with a step that identifies whether an event type is reported (at least once, anywhere) in a document. We argue for treating this as a document classification task.
2. **Event attribute identification**: We conceptualize political events as having four generic *attributes*, the event's *actor*, *recipient*, *date*, and *location*. We pose this as an information extraction problem and introduce a small fine-tuned language model to extract the corresponding spans from a document.
3. **Entity Resolution**: named political entities, such as people or organizations, are easier to analyze and categorize if they can be linked to a canonical external knowledge base. We introduce a new model for linking extracted names to their correct article in an offline copy of Wikipedia.
4. **Entity Categorization**: political entities, whether generic or named entities resolved to Wikipedia, are categorized according to the researcher's ontology (e.g., that "police" and "internal security forces" are both members of a broader LAW ENFORCEMENT category.)
5. **Date and Location Resolution**: events' relative dates ("last Wednesday") are resolved to absolute dates and extracted event locations ("Aleppo") are resolved to their geographic coordinates. Specifically, we resolve place names to their entries in the Geonames gazetteer (Wick and Boutreux, 2011) to obtain their geographic coordinates.

Researchers can use pretrained models for each step, customize only the components relevant to their project, or employ individual steps to solve non-event data problems.

3.1 LLMs and the "Test of Time"

The rapidly improving capabilities of LLMs raise a natural question: why build specialized tools rather than using large generative LLMs to produce event data end-to-end (Sutton, 2019)? We offer three responses. First, resolving actors and locations to canonical identifiers such as Wikipedia or Geonames (Steps 3 and 5), requires querying canonical knowledge stores rather than relying on the pretrained knowledge in LLM weights. Second, processing entire corpora through large LLMs imposes major computational costs, and using closed-weight models has well-established reproducibility concerns (Palmer, Smith and Spirling, 2024). SI E.2 provides an end-to-end comparison of our pipeline on a new application to a frontier LLM, showing that while frontier LLMs outperform our attribute model on recipients (but not actor recognition), they come with clear replicability and monetary costs. Third, our pipeline's modularity means that as LLMs improve, researchers can indeed

use them for individual steps in the pipeline, such as document classification or the attribute model, without replacing the full system.

4 Customizing Event Data

Before describing the technical details of each step, we outline the research design decisions researchers face when producing custom event data. The simplest scenario—a researcher coding new event types from English-language news text—requires customizing only the event detection step, while the pretrained attribute, entity linking, and resolution models can be used without modification. More substantial departures require progressively more effort, as summarized in Table 1.

Event types. Researchers must decide which event types to classify, at what level of granularity, and how each event type is defined (Halterman and Keith, 2026). For example, distinguishing hunger strikes from demonstrations provides greater granularity about protest tactics, but requires more annotation effort. Political scientists' familiarity with document classification, combined with techniques such as synthetic data augmentation (Halterman, 2025) and fine-tuning (Hu et al., 2022), makes training new event classifiers feasible.

Event attributes. Defining new event types changes the inputs to the attribute model, which must extract attributes for event types it was not trained on. In most cases, the pretrained model extracts the core four attributes even for unseen event types, as we demonstrate in our validations below. Researchers who need additional attributes (e.g., "number of participants" in a protest) may need to generate new training data, but our synthetic data pipeline reduces the effort required.

Actor categories. Researchers using our pipeline define actor categories by editing a human-readable text file of role descriptions that are matched via semantic similarity. Modifying the actor categorization only requires updating the mapping file with patterns in the relevant categories and does not require retraining any models.

Text sources. Our pipeline is designed for English-language news text. Stylistically similar text should work without modification, though very different domains may require additional training data. Non-English languages would require translation or significant retraining (see Sections 7 and SI H).

We next describe new techniques for doing each of these steps. We address the annotation burden through synthetic training data, the sentence-level limitation through document-level extraction, and the actor dictionary problem through Wikipedia-based entity linking. The result is an end-to-end system that can produce custom event data without requiring dictionary development or computational linguistics expertise, and also a modular set of models and techniques that can be used independently.

Customization	Pipeline Step Affected	Customization Required
New event types	Event detection	Train classifiers on labeled (synthetic) examples
Event subtypes/modes	Event detection	Train second-stage classifiers
Custom attributes	Attribute extraction	Extend (synthetic) training data and retrain model
Actor category ontology	Entity categorization	Edit category mapping file
Whether to link named entities	Entity linking	Optional; no changes needed
Non-English text	All steps	Retrain models; new Wikipedia index

Table 1: *Mapping customization decisions to pipeline steps and required effort.*

4.1 Step 1: Event Detection

The first step in our pipeline is event detection: identifying which events are reported in a document. We suggest that researchers treat this as a document classification task, following an increasingly common approach in the literature (Hu et al., 2022; Zhukov and Ayers, 2023; Halterman and Keith, 2025). In addition to the benefits discussed in Section 2, treating event detection as document classification makes this step easy to customize. Political scientists have access to several techniques for making document classification more efficient, including through zero-shot classification techniques (Halterman et al., 2021; Burnham et al., 2025), synthetic data augmentation (Halterman, 2025), and active learning-based annotation (Miller, Linder and Mebane, 2020). In particular, pretrained models such as ConfliBERT (Hu et al., 2022; Brandt et al., 2025) can be efficiently fine-tuned to produce an accurate event classifier.

A clear way to pose the problem is as a set of binary event classifiers. This also greatly eases the annotation process: rather than completely annotating each document for all event types, researchers can collect binary annotations of each event type for documents.[6]

Because event detection as document classification is well established, we

[6]Researchers can also use document classifiers to identify broader themes or "contexts" of a document or nested event types: sub-types such as "hunger-strike" or "demonstration" within a broader PROTEST category can be identified with a second-stage classifier. See SI A. Similarly, classifiers could be trained to distinguish more clearly between actual, planned, hypothetical, or historical events.

do not introduce new models for this step. Our technical contributions begin with the subsequent steps in the pipeline, starting with attribute extraction, which also addresses limitations of document-level classification such as localizing and distinguishing multiple events within a single story.

4.2 Step 2: Extracting Event Attributes

After identifying that a document contains a set of events, the next step in our conceptual framework is to identify the *attributes* of each event. We pose this as an *extraction* problem: for each event of a given type that the document reports, the model should return the spans of text that report information on the participants, date, and location of the event.

In contrast to the computational linguistics literature, which defines different attributes or "arguments" for each event type, we deliberately adopt a generic set of event attributes. For example, we treat the *victim* of an attack and the *beneficiary* of aid as equivalent *recipients*. We focus on four core attributes shared by most political event types:

- **Actor**: Which person, organization, or entity is initiating or carrying out the event?
- **Recipient**: Which entity is target, victim, beneficiary, or recipient of the event?
- **Location**: Where did the event take place?
- **Date**: When did the event take place?

Some event types may not have all attributes (e.g., a bilateral meeting may have only actors and no recipients), some attributes may not be reported in the text, and some attributes may have multiple answer spans. A single document may also report several events of the same type (e.g., protests in three different cities), each with its own attributes.

One approach from the prior literature is to frame attribute extraction as a question-answering task, posing natural language questions to a pretrained question-answering (QA) model for each attribute and event type (Du and Cardie, 2020). However, this approach requires separate questions for each event-attribute combination, asks each question in isolation (which can lead to "role flipping" where actor and recipient are confused), and struggles with absent or multi-part answers. Prior work found that only 34.5% of event attributes could be identified from the sentence describing the event (Li, Ji and Han, 2021), which motivates our document-level extraction.

4.2.1 A New Model for Event Attribute Extraction

We introduce a single model that extracts attributes for arbitrary event types from English-language text, trained on public domain news text and synthetic stories, and small enough to run on consumer hardware.

We fine-tune a small generative LLM that takes as input a (document, event description) tuple and yields a structured JSON list with the actor, recipient, date, and location spans, along with an "anchor quote" from the text that identifies which event the record describes, allowing the attribute model to identify multiple instances of the same event type in a single document. Generating all attributes jointly in a single pass avoids the role flipping and per-attribute overhead of QA approaches, and allows us to use the capabilities and inference efficiency of modern LLMs.

Large LLMs are capable of performing this task (Salau and Halterman, 2025) and may be the right approach for some projects, given their ease of use. However, commercial closed-weight LLMs carry costs: they are not reproducible (Palmer, Smith and Spirling, 2024), and large open-weight LLMs (10B+ parameters) are costly to run at large scale or have prohibitive hardware requirements. We therefore adopt a distillation approach to train a small model: we use a larger "teacher" LLM to generate training data (Halterman, 2025), then fine-tune a much smaller "student" model, Qwen3.5-0.8B (800M parameters), on this data.

We build the training data in two steps (see SI B.1). First, we assemble documents from two sources: 884 real public-domain news articles from Voice of America (VOA), and 1,008 synthetic news stories generated by open weight models.[7] The synthetic stories contain specified event types, and we vary the prompted country, document length, writing style, event placement, and completeness of reporting to produce a diverse training set. The event types and descriptions are taken from the PLOVER event ontology (Halterman et al., 2023). Second, we provide a separate larger LLM with each document, the event type definition, and detailed attribute extraction rules based on standard practice in natural language processing (SI E.4), and ask it to return every instance of the event type in JSON format. Because most pairs contain no event of the given type, the model learns to return an empty list when the event is absent. A benefit of this approach is that we can generate a large and diverse set of training examples cheaply and release the training data in its entirety, which would not be possible with annotated copyrighted news text.

Because the fine-tuned model's output is conditioned on event type definitions provided at inference time, researchers can adapt it to entirely new event types simply by writing a new definition without any additional annotation or retraining. When we removed two event types from the training data entirely and provided only their definitions, the model, seeing for the first time, extracted their attributes about as accurately as a model trained on them (SI B.4). We evaluate its performance in Section 5.

[7] We use different LLMs families the two stages, Gemma3-12B and Gemma-4-31B for story generation and GLM 5.3 flash for attribute labeling, to avoid concerns about over-reliance on a single idiosyncratic model.

4.3 Step 3: Linking Entities

For applied users of event data, the raw spans of text describing an event's actors and recipients need further processing to be useful. Specifically, named entities such as people or organizations should be disambiguated and resolved to a unique identifier. For example, a researcher would want a model to identify, from context, whether "Bush" and "George Bush" refer to the 41st or 43rd US president. Prior dictionary-based methods are limited here, requiring exhaustive entity lists and exact string matches.

We develop a new model to link mentions of political entities with their correct Wikipedia entries, using the context of the surrounding news document. Resolving each extracted entity to Wikipedia provides both a unique, canonical identifier for each entity, and information to correctly categorize the entity in the following step.

4.4 A New Model for Linking Entities to Their Wikipedia Pages

We train and validate a method that resolves mentions of named entities to their English Wikipedia pages, drawing on the context of the news text to select the best article. The method consists of two broad steps. First, we query an offline Wikipedia index with the entity's name to obtain a list of candidate articles. Second, we use a supervised model to select the best article from among the candidates.

To query Wikipedia, we build an offline Elasticsearch index from a complete dump of English-language Wikipedia.[8] The index recreates Wikipedia's redirect network (e.g., "ISIS" → "Islamic State") and extracts alternative names from each article's first paragraph, both of which are valuable for resolving ambiguous names. Details on processing and ingestion are in SI C.1. The Elasticsearch query searches for exact and fuzzy matches on article titles, redirects, and alternative names, with lower-weighted searches on introduction paragraphs and short descriptions. The correct article appears in the top 200 results 95.7% of the time.[9]

We then train an XGBoost classifier to select the correct article from among the candidates, or to decline to link when none is correct. The classifier uses exact and fuzzy string matching, semantic similarity between the mention's document context and candidate article content, role and country matching, and proxies for the article's importance. The full feature list and an ablation study are in SI C.3. We train the classifier on 1,592 hand-annotated entity-article pairs from public domain *Voice of America* articles and evaluate on 374

[8] Researchers can download and use our pre-built index or rebuild the database in ∼2 hours using scripts we provide.

[9] Reducing to 50 candidates lowers recall to 92.5%.

mentions from 59 held-out articles.[10] When we evaluate the model on held-out test documents, it achieves 87.7% accuracy.

Because Wikipedia is continuously updated and more comprehensive than any individual research project could be, researchers can keep their entity resolution current by periodically downloading a new Wikipedia dump rather than manually maintaining actor dictionaries. This addresses the primary obstacle for maintaining dictionary-based event coders.

Once the correct article is identified, we parse out structured information from its information box and introductory paragraph, including office titles, dates in office, and short descriptions, to provide inputs for the entity categorization step. Details on this extraction are in SI C.6.

4.5 Step 4: Entity Categorization–Assigning Ontology Categories to Actors and Recipients

Resolving an entity to Wikipedia provides a canonical identifier, but researchers typically also need to categorize entities into substantively meaningful groups defined by their substantive questions. For instance, a researcher studying civil conflict might want to categorize "Syrian government forces," "President Assad," and "government troops" under a single `SYR GOV` code, while distinguishing them from "Syrian rebels" and "Free Syrian Army" (`SYR REB`). Similarly, a researcher studying labor politics might need to identify all mentions of "union leaders," "labor organizers," and specific union officials as belonging to a custom `LABOR` category of entities.

Our approach handles both named entities linked to Wikipedia (e.g., "Antony Blinken") and generic actor mentions (e.g., "Latvian air force," "protesters"). We represent each category in the researcher's ontology as a set of text patterns describing typical roles for that category. For example, the GOV category includes patterns like "government official" and "president," while MIL includes "armed forces" and "military officer." These patterns are stored in a human-readable text file that researchers can easily modify to define their own ontologies.[11] We encode both the entity description and all ontology patterns using a sentence transformer and select the highest-similarity match above a threshold. Unlike dictionary-based approaches, this does not require exact matches between text spans and category entries. Details are in SI D.

Defining a new actor ontology thus requires only editing a text file of category descriptions, not exhaustively enumerating every possible actor mention.

[10] Details on the gold standard dataset are in SI C.4.

[11] Our default ontology is based on the PLOVER coding scheme (Halterman et al., 2023), but researchers can substitute their own.

4.6 Step 5: Dates and Locations

Finally, the extracted information about the event's date and location need to be resolved to canonical references. Assuming that a story's events occur on the story's date of publication can lead to serious inferential problems. Instead, we resolve the extracted date span to an absolute date (or date ranges) in reference to the article's publication date (with reported day, week, month, or year granularity), by using the existing Python `dateparser` library, and a set of custom rules to handle common patterns such as "last Monday" or "over the weekend."

To identify the event's location, we employ the existing Mordecai3 geoparsing library (Halterman, 2023). Mordecai3 extracts all locations in a document and resolves them to their entries in the Geonames gazetteer, which provides the geographic coordinates of each location. We compare the extracted location span from the attribute model to each of the geoparsed locations and identify any spans with overlap.

5 Evaluation

We evaluate the pipeline's components on internal held-out data and expert annotations, and then validate the full pipeline's transferability on an external dataset with a completely different ontology.

5.1 Internal Validation of the Attribute Model

An accurate attribute model will correctly extract the spans of text that correspond to a provided event's attributes. We evaluate the attribute model against two separate sets of annotated public domain *Voice of America* articles. First, we perform evaluation on a small set of 100 *Voice of America* articles, each independently coded by three authors. This evaluation annotates only PLOVER ASSAULT events, and only the first ASSAULT event in each article. As a result, we cannot use it to estimate the attribute model's precision, nor its performance on other event categories. The results (Table 2) show that among the assault events identified by the attribute model, its spans exactly match the gold evaluation set for between 55.9% (recipients) and 98.5% (dates). Allowing partial overlap of the spans yields accuracies above 80% for all spans.

To evaluate the model on a broader set of events and completely annotated events, we produce an evaluation set of 500 *Voice of America* documents labeled for all PLOVER event types, under an annotation scheme built on standard definitions in natural language processing (SI E.4). The annotation scheme provides clear instructions about the correct extent of a span and how to pick the most informative "canonical" span among several mentions of the same entity. Three open-weight LLMs drafted each unit independently,

Field	Exact	Overlap
Actor	67.6	81.2
Recipient	55.9	90.4
Location	69.1	91.2
Date	98.5	98.5
Coded attacks found (of 93)	68	

Table 2: *Attribute model accuracy on a sample of 100 hand-labeled stories for ASSAULT events. Note that the spans are evaluated only if the event is detected (68 of 93 gold events).*

with a human adjudicator evaluating documents with high disagreement between models and agreeing with the resulting annotations around 75% of the time. This yields 1,048 events.

Table 3 reports the attribute model's performance on this evaluation set. The attribute model is conservative in its event detection: it refrains from identifying events more than it invents events. Its missed events are concentrated in articles that report several events of the same type: event recall is 88% in articles with one event and 67% in articles with two or more. We illustrate this by decomposing the attribute's model accuracy into performance on all events and only on the subset of events it identifies. Our standard measure allows for partial overlap between gold and predicted spans (e.g., "police" and "police officers" would match). We find that among events that the attribute model detects, F1 scores are between 65.0 and 77.9, with recipients being the most difficult challenge. A stricter test, requiring word-for-word matches to the gold spans reduces actor and recipient accuracy by around 10–15 points. These scores are comparable to those reported in the NLP literature (SI G).

The model also extracts attributes for event types it never saw in training when it is given only their definitions (SI B.4). It also rarely confuses actors and recipients: only 1.0% of the key's actor and recipient spans appear in the other role in the model's output. The model processes approximately 36 events (18 documents) per second on a consumer GPU[12] and can run on CPUs at a tolerable speed, around about 2-4 seconds per event.

5.2 Internal Evaluation of the Wikipedia Model

For entity linking, our Wikipedia candidate retrieval returns the correct article in 95.7% of queries when considering up to 200 candidates. The supervised XGBoost classifier then selects the correct article, or correctly declines to link a mention that has no page, with 87.7% accuracy on held-out test data

[12]Nvidia RTX 4090.

	All events			Found events			
Role	Precision	Recall	F1	Precision	Recall	F1	Exact F1
Actor	53.7	51.7	52.7	70.9	72.5	71.7	60.8
Recipient	53.5	47.1	50.1	66.3	63.7	65.0	51.7
Date	62.5	55.6	58.9	80.7	75.3	77.9	76.7
Location	54.7	54.9	54.8	70.9	73.7	72.2	68.5
Mean of four roles	56.1	52.3	54.1	72.2	71.3	71.7	64.4
Events found	76.5	71.1	73.7				

Table 3: *Performance of the fine-tuned attribute model on the held out evaluation set of 500 PLOVER-annotated documents. The model is conservative in identifying events (left columns, missed events are 0s for all attributes), but more accurate on detected events. All results allow partial span overlap except the final column.*

(see SI C.4). The model is precise (it links to an article, 92.2% are correct) and has high recall (88.6% of the mentions that have a page).

6 Cross-Ontology Validation: Electoral Contention and Violence Dataset

To validate the pipeline's ability to generate custom event data, we employ it to replicate hand-coded event data from the Electoral Contention and Violence dataset (ECAV) (Daxecker, Amicarelli and Jung, 2019), which codes violence, intimidation, and protests related to elections. ECAV's event types and actor categories differ entirely from the PLOVER ontology we used in developing the pipeline. We retrain none of the pipeline's components, only rewriting event type definitions and a file that maps generic entity mentions to ECAV categories. We test every step of the pipeline after event detection: attribute extraction, actor resolution and categorization, geocoding, and date resolution. Performance on this dataset represents realistic accuracy for a researcher using our pipeline on a new domain without retraining any components.

Among event datasets, ECAV has two unique features that make it useful for validation: each record includes the raw spans that annotators recorded for actors and targets and enough metadata to link the event to its original news story.

We select the seven elections with the greatest number of ECAV events (SI E). We link each event to its original news article by matching the source, date, and title metadata to a LexisNexis corpus, which we are able to do for

46%–84% of events per election. After additional filtering (SI E), we arrive at 1,662 events.

For each record, the attribute model (Step 2) receives the full article, ECAV's selected passage for the event, and the event and its definition. The extracted spans then pass through actor resolution (Step 3), categorization (Step 4), and geocoding and date resolution (Step 5). For entity categorization, we write a new custom dictionary that maps generic role descriptions ("supporters", "police", "gunmen") to ECAV's five actor categories.

6.1 ECAV Evaluation

Comparing the model's spans directly with ECAV's is complicated because ECAV often reports non-verbatim text, which our attribute model disallows, or makes assumptions about the targets of protests ("Government", if not otherwise specified). Allowing for partial matches, 45.4% of actor spans, 22.9% of recipient spans, and 40.1% of locations match ECAV's (SI E).

A more realistic evaluation is an end-to-end comparison: how often does NGEC's pipeline match the entity categories and locations of ECAV? Table 4 reports the share of events where the pipeline's final value matches ECAV's. "ECAV spans" feeds ECAV's own spans into the steps after extraction; errors here result from dictionary mismatches and some inconsistencies in ECAV itself (E.3). "Raw NGEC" uses our attribute spans and our entity resolver. The "+ rules" column deterministically applies a small number of corrections to match ECAV (e.g., "Government" as the recipient of protests if not otherwise stated.)

Field	ECAV spans	Raw NGEC	NGEC + rules
Actor category	88.9	62.3	63.2
Recipient category	89.6	45.2	55.2
Location ($\leq$25 km)	71.8	54.8	54.8
Date (exact)	—	85.8	85.8
Both right (actor, recipient)	80.2	30.4	37.3
Actor, recip., location right	57.5	20.0	22.5

Table 4: *End-to-end accuracy on the 1,662 ECAV events: percent of events where the pipeline's final value matches ECAV's coded value. "ECAV spans" uses ECAV's own hand-coded names in the steps after extraction. "Raw NGEC" uses our attribute model's spans, with the pipeline exactly as it runs. "NGEC + rules" adds two ECAV coding conventions: a protest, strike, blockade, or riot with no target gets the target State, and an arrest with no actor gets the actor State.*

Most remaining errors are omissions or near misses (SI E.3). SI E.2 re-

ports a comparison with three off-the-shelf LLMs, finding that they outperform NGEC on end-to-end recipient and location extraction, but not actors, and with replicability and cost implications (Tables 10, 11).

7 Limitations and Future Work

Our pipeline is currently limited to English-language text. Extending to non-English contexts would require retraining the attribute model, rebuilding the Wikipedia index, and reconfiguring the similarity-based components. Whether this is preferable to applying our English-language pipeline to machine-translated text remains an open question (Osorio et al., 2025).

Second, specific components of the pipeline can be improved. For example, our ECAV validation suggested that integrating story-level context into the entity categorization step and not just the Wikipedia lookup step could improve actor resolution. More broadly, cascading errors (and propagating uncertainty) across pipeline components remain a well-known problem for sequential data processing pipelines that we measure, but do not fully address here.

Third, our pipeline does not address several longstanding event data challenges, including duplicate extraction of the same event from multiple sources, resolving conflicting information from across multiple reports, and media reporting biases that cause systematic under-coverage of certain event types and areas.

8 Conclusion

This paper introduces a modular framework and open-source pipeline that lowers the barriers to producing custom event data. Our five-part decomposition—event detection, attribute extraction, entity linking, entity categorization, and date/location resolution—clarifies the decisions that researchers face when coding custom event data with their own ontologies and allows them to customize only the components their project requires. To address the two most challenging steps, we develop a distilled language model for attribute extraction that can generalize to new event types without hand annotation, and a Wikipedia-based entity linking system that disambiguates political actors without hand-built dictionaries. Our validation against expert annotations and the ECAV dataset—using a completely different ontology and no retraining—shows that the attribute model transfers to new domains. The lower performance of attribute extraction suggests that researchers should focus their customization efforts here. Because the pipeline runs on consumer hardware, relies only on open-weight models, and requires no proprietary APIs,

it is accessible to researchers regardless of their computational resources or institutional setting. We hope that by making custom event data production faster and cheaper, this work will encourage researchers to build event datasets that are well matched to their substantive questions rather than accepting the measurement tradeoffs inherent in repurposing existing datasets.

9 Funding

This research was sponsored by the U.S. Government. The views expressed in this paper are the authors' alone and do not represent the views of the U.S. Government.

Funding for PLOVER was initially provided in part by the U.S. National Science Foundation award SBE-1539302, "RIDIR: Modernizing Political Event Data for Big Data Social Science Research".

10 Acknowledgments

We gratefully acknowledge the work done by colleagues at Leidos: John Berrie, Heather Carter, Kathleen Krupenevich, and Jevon Spivey. Emma Huizenga, Gayathri Emuru, and Rakesh Emuru provided excellent research assistance. Benjamin Radford, Leah Windsor, and Michael Colaresi, along with attendees at ISA, APSA, and PaCSS, provided helpful comments. Students in the ICPSR 2025 summer workshop on event extraction provided helpful feedback on the NGEC coding pipeline. We acknowledge the use of Claude Code, primarily Opus 5 and Fable 5, in assisting with writing code for the NGEC pipeline and preparing replication materials.

References

Adcock, Robert and David Collier. 2001. "Measurement validity: A shared standard for qualitative and quantitative research." *American Political Science Review* 95(03):529–546.

Ahn, David. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*. pp. 1–8.

Alsarra, Sultan, Mubarak Alrashoud, Javier Osorio, Vito D'Orazio, Latifur Khan and Patrick T Brandt. 2025. "Multilingual approaches to extractive question answering in political texts." *Computational and Mathematical Organization Theory* 31(4):299–322.

Althaus, Scott, Joseph Bajjalieh, John F. Carter, Buddy Peyton and Dan A. Shalmon. 2020. "Cline Center Historical Phoenix Event Data.".

Baker, Collin F, Charles J Fillmore and John B Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics pp. 86–90.

Boschee, Elizabeth, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz and Michael Ward. 2015. "ICEWS Coded Event Data.".
URL: *https://doi.org/10.7910/DVN/28075*

Brandt, Patrick T, Sultan Alsarra, Vito D'Orazio, Dagmar Heintze, Latifur Khan, Shreyas Meher, Javier Osorio and Marcus Sianan. 2025. "Extractive versus Generative Language Models for Political Conflict Text Classification." *Political Analysis* pp. 1–29.

Brathwaite, Robert and Baekkwan Park. 2018. "Measurement and Conceptual Approaches to Religious Violence: The Use of Natural Language Processing to Generate Religious Violence Event-Data." *Politics and Religion* pp. 1–42.

Burnham, Michael, Kayla Kahn, Ryan Yank Wang and Rachel X Peng. 2025. "Political DEBATE: Efficient zero-shot and few-shot classifiers for political text." *Political Analysis* .

Dai, Yaoyao, Benjamin Radford and Andrew Halterman. 2022. "Political Event Coding as Text to Text Sequence Generation." *Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2022)* .

Das, Dipanjan, Desai Chen, André FT Martins, Nathan Schneider and Noah A Smith. 2014. "Frame-semantic parsing." *Computational linguistics* 40(1):9–56.

Daxecker, Ursula, Elio Amicarelli and Alexander Jung. 2019. "Electoral Contention and Violence (ECAV): A new dataset." *Journal of Peace Research* 56(5):714–723.

Doddington, George R, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel and Ralph M Weischedel. 2004. The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation. In *LREC*. Vol. 2 p. 1.

Du, Xinya and Claire Cardie. 2020. "Event Extraction by Answering (Almost) Natural Questions." *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* pp. 671–683.

Ebner, Seth, Patrick Xia, Ryan Culkin, Kyle Rawlins and Benjamin Van Durme. 2020. "Multi-Sentence Argument Linking." *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* pp. 8057–8077.

Gerner, Deborah J., Philip A Schrodt, Omür Yilmaz and Rajaa Abu-Jabr. 2002. "Conflict and Mediation Event Observations (CAMEO): A new event data framework for the analysis of foreign policy interactions." *International Studies Association, New Orleans* .

Getman, Jeremy, Joe Ellis, Zhiyi Song, Jennifer Tracey and Stephanie M Strassel. 2017. Overview of Linguistic Resources for the TAC KBP 2017 Evaluations: Methodologies and Results. In *TAC*.

Goertz, Gary. 2020. *Social science concepts and measurement: New and completely revised edition*. Princeton University Press.

Halterman, Andrew. 2023. "Mordecai3: A Neural Geoparser and Event Geocoder." *arXiv preprint arXiv:2303.13675* .

Halterman, Andrew. 2025. "Synthetically generated text for supervised text analysis." *Political Analysis* 33(3):181–194.

Halterman, Andrew, Benjamin E Bagozzi, Andreas Beger, Philip A Schrodt and Grace I Scarborough. 2023. "PLOVER and POLECAT: A New Political Event Ontology and Dataset." *International Studies Association Conference Paper* .

Halterman, Andrew, Jill Irvine, Christan Grant, KT Jabr and Yan Liang. 2018. Creating an automated event data system for Arabic text. In *ISA Annual Meeting 2018 in San Francisco*.

Halterman, Andrew and Katherine A Keith. 2025. "Codebook LLMs: Evaluating LLMs as Measurement Tools for Political Science Concepts." *Political Analysis* .

Halterman, Andrew and Katherine A Keith. 2026. "What is a protest anyway? Codebook conceptualization is still a first-order concern in LLM-era classification." *Association for Computational Linguistics: Main Papers* .

Halterman, Andrew, Katherine A Keith, Sheikh Muhammad Sarwar and Brendan O'Connor. 2021. "Corpus-Level Evaluation for Event QA: The IndiaPoliceEvents Corpus Covering the 2002 Gujarat Violence." *Findings of the Association for Computational Linguistics* .

Haunss, Sebastian, Priska Daphi, Jan Matti Dollbaum, Lidiya Hristova, Pál Susánszky and Elias Steinhilper. 2025. "PAPEA: A modular pipeline for the automation of protest event analysis." *Political Science Research and Methods* pp. 1–18.

Hsu, I-Hung, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang and Nanyun Peng. 2022. DEGREE: A Data-Efficient Generation-Based Event Extraction Model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pp. 1890–1908.

Hu, Yibo, Mohammad Saleh Hosseini, Erick Skorupa Parolin, Javier Osorio, Latifur Khan, Patrick Brandt and Vito D'Orazio. 2022. ConfliBERT: A Pretrained Language Model for Political Conflict and Violence. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pp. 5469–5482.

Ji, Heng and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: Hlt*. pp. 254–262.

Leetaru, Kalev and Philip A Schrodt. 2013. GDELT: Global data on events, location, and tone. In *ISA Annual Convention*.

Li, Sha, Heng Ji and Jiawei Han. 2021. "Document-level event argument extraction by conditional generation." *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pp. 894–908.

Liang, Yan, Khaled Jabr, Christan Grant, Jill Irvine and Andrew Halterman. 2018. New techniques for coding political events across languages. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*. IEEE pp. 88–93.

Lin, Ying, Heng Ji, Fei Huang and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th annual meeting of the association for computational linguistics*. pp. 7999–8009.

Meher, Shreyas and Patrick T Brandt. 2025. "ConflLlama: Domain-specific adaptation of large language models for conflict event classification." *Research & Politics* 12(3):20531680251356282.

Mihalcea, Rada and Andras Csomai. 2007. Wikify! Linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. pp. 233–242.

Miller, Blake, Fridolin Linder and Walter R Mebane. 2020. "Active Learning Approaches for Labeling Text: Review and Assessment of the Performance of Active Learning Approaches." *Political Analysis* pp. 1–20.

Norris, Clayton, Philip Schrodt and John Beieler. 2017. "PETRARCH2: Another event coding program." *The Journal of Open Source Software* 2(9).

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

Osorio, Javier, Afraa Alshammari, Naif Alatrush, Dagmar Heintze, Amber Converse, Sultan Alsarra, Latifur Khan, Patrick T. Brandt and Vito D'Orazio. 2025. The Devil is in the Details: Assessing the Effects of Machine-Translation on LLM Performance in Domain-Specific Texts. In *Proceedings of Machine Translation Summit XX: Volume 1*. Geneva, Switzerland: European Association for Machine Translation pp. 315–332.
URL: *https://aclanthology.org/2025.mtsummit-1.24/*

Osorio, Javier and Alejandro Reyes. 2017. "Supervised Event Coding From Text Written in Spanish: Introducing Eventus ID." *Social Science Computer Review* 35(3):406–416.

Palmer, Alexis, Noah A Smith and Arthur Spirling. 2024. "Using proprietary language models in academic research requires explicit justification." *Nature Computational Science* 4(1):2–3.

Palmer, Glenn, Roseanne W. McManus, Vito D'Orazio, Michael R. Kenwick, Mikaela Karstens, Chase Bloch, Nick Dietrich, Kayla Kahn, Kellan Ritter and Michael J. Soules. 2022. "The MID5 Dataset, 2011–2014: Procedures, coding rules, and description." *Conflict Management and Peace Science* 39(4):470–482.

Palmer, Martha, Daniel Gildea and Paul Kingsbury. 2005. "The proposition bank: An annotated corpus of semantic roles." *Computational linguistics* 31(1):71–106.

Peng, Hao, Xiaozhi Wang, Feng Yao, Kaisheng Zeng, Lei Hou, Juanzi Li, Zhiyuan Liu and Weixing Shen. 2023. The devil is in the details: On the pitfalls of event extraction evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*. pp. 9206–9227.

Radford, Benjamin J. 2019. "Automated dictionary generation for political event coding." *Political Science Research and Methods* pp. 1–15.

Raleigh, Clionadh, Roudabeh Kishi and Andrew Linke. 2023. "Political instability patterns are obscured by conflict dataset scope conditions, sources, and coding choices." *Humanities and Social Sciences Communications* 74(10).

Ratinov, Lev, Dan Roth, Doug Downey and Mike Anderson. 2011. Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*. pp. 1375–1384.

Salau, Tosin and Andrew Halterman. 2025. "Reprisals Against Civilians in Nigeria: Understanding Events in Sequence." *PolMeth conference paper* .

Scarborough, Grace I, Benjamin E Bagozzi, Andreas Beger, John Berrie, Andrew Halterman, Philip A Schrodt and Jevon Spivey. 2023. "POLECAT weekly data." *Harvard Dataverse* 62:2023.

Schrodt, Philip A. 2009. "TABARI: Textual analysis by augmented replacement instructions." *Dept. of Political Science, University of Kansas, Blake Hall, Version 0.7. 3B3* pp. 1–137.

Schrodt, Philip A, Shannon G Davis and Judith L Weddle. 1994. "Political science: KEDS—a program for the machine coding of event data." *Social Science Computer Review* 12(4):561–587.

Sundberg, Ralph and Erik Melander. 2013. "Introducing the UCDP Georeferenced Event Dataset." *Journal of Peace Research* 50(4):523–532.

Sutton, Richard S. 2019. "The Bitter Lesson." Incomplete Ideas (blog). Accessed: 2026-03-02.
URL: *http://www.incompleteideas.net/IncIdeas/BitterLesson.html*

Toft, Monica Duffy and Yuri M Zhukov. 2015. "Islamists and Nationalists: Rebel Motivation and Counterinsurgency in Russia's North Caucasus." *American Political Science Review* 109(02):222–238.

Wadden, David, Ulme Wennberg, Yi Luan and Hannaneh Hajishirzi. 2019. Entity, Relation, and Event Extraction with Contextualized Span Representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics.

Walker, Christopher, Stephanie Strassel, Julie Medero and Kazuaki Maeda. 2006. "ACE 2005 multilingual training corpus." *Linguistic Data Consortium* .

Wick, Marc and C Boutreux. 2011. "GeoNames." *GeoNames Geographical Database* .

Zhukov, Yuri and Natalie Ayers. 2023. "VIINA 2.0: Violent Incident Information from News Articles on the 2022 Russian Invasion of Ukraine.".
URL: *https://github.com/zhukovyuri/VIINA*