Massachusetts Institute of Technology
Political Science Department

Research Paper No. 2018-21

Linking Events and Locations in Political Text

Andrew Halterman, Massachusetts Institute of Technology

Electronic copy available at: https://ssrn.com/abstract=3267476

Linking Events and Locations in Political Text

ANDREW HALTERMAN

This work proposes the first general technique for automatically geolocating political events in text as one step in a broader project of recognizing political relationship and extracting political information from text. Using techniques borrowed from computational linguistics and a novel set of labeled sentences, I create an method to link events and locations in text that triples the accuracy of a reasonable baseline model. I describe the potential uses of such a system in political science, describe the neural-net based model, and demonstrate its ability to answer an open question on the role of conventional military offensives in causing civilian casualties in the Syrian civil war.

Introduction

Research in much of political science, and especially in comparative politics and security studies, has been undergoing a turn toward micro data in the past 15 years.[1] Subnational variation at very fine resolutions has become one of the major

[1]For valuable feedback on earlier versions of this paper, I think John Beieler, Fotini Christia, In Song Kim, Rich Nielsen, and Rachel Tecott. Emily Young provided excellent assistance in annotating several thousand sentences as training data. I gratefully acknowledge the support of a National Science Foundation Graduate Research Fellowship. For support in developing Mordecai (https://github.com/openeventdata/mordecai) and for creating annotated text data, I thank the National Science Foundation under award number SBE-SMA-1539302, the Defense Advanced Research Project Agency's XDATA program, and the U.S. Army Research Laboratory and the U.S. Army Research Office through the Minerva Initiative under grant number W911NF-13-0332. Any opinions, findings, and conclusions or recommendations expressed in this material are those

Electronic copy available at: https://ssrn.com/abstract=3267476

source of empirical puzzles and evidence in these two subfields. At the same time, automated text analysis is becoming one of the most important sources of new data in the field. Existing methods for text analysis have produced great insight into survey responses, elite discourse, and legislative priorities, but has only just begun to contribute to more micro-level studies in political science. I develop and describe a method that enables researchers to link these two trends, allowing them to easily link events coded from text to the specific locations where they are reported to occur.

I use the term "event geoparsing" to describe the process of taking text with already recognized events and producing geographic coordinates for the actions or events described in it. Specifically, I provide a system that, given a sentence and a verb of interest in the sentence, will return the place names from the sentence where the verb took place. Formulated as a general task, this is a novel problem in both political science and computer science. The task is made difficult by the problems of recognizing events and locations in text in the first place, and is further impeded by sentences with multiple events and multiple locations. A baseline model that makes the reasonable assumption that events (specifically defined as a verb) should be assigned to the closest location word in the text achieves an F1 accuracy score for word classification of 0.27.[2] Drawing on a set of almost 10,000 hand-labeled sentences, I train a recurrent neural net that draws on a rich set of linguistic features to label a sequence of text with labels for whether the word is a location word corresponding to a specified verb. Measured by token, the model produces an F1 score of 0.82 and a simple accuracy measure of over 99%.

The ability to provide fine-grained geolocated data from text opens new possibilities for testing theories and answering important open questions in political science. I demonstrate how this technique can address the literature on violence in civil wars by producing a dataset of 1,760 military offensives in Syria in 2016,

of the author and do not necessarily reflect the views of the National Science Foundation or the Department of Defense.

[2]I use the accuracy metric that is standard for this kind of task in computer science, the F1 score. The F1 score is the harmonic mean of the model's precision and recall, where precision is the proportion of the positive labels the model returns that are indeed true positives, and recall is the proportion of true positives that the model recovers. High recall means the model correctly identities most true positives, and precision means that it also generates few false positives. The F1 score is a balance of the two: $F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$ and a perfect F1 score would be 1.

Electronic copy available at: https://ssrn.com/abstract=3267476

with precise coordinates to the finest resolution reported in the text. This dataset is important in understanding the extent to which violence against civilians in Syria is the product of insurgent- or unconventional violence (Kalyvas 2006; Weinstein 2006) or the product of conventional military action along defined front lines (Balcells and Kalyvas 2014; Balcells 2017). Using the dataset, I am able to measure the number of civilian casualties occurring in the vicinity of military offensives, giving us greater insight into the causes of civilian victimization in Syria. Finally, I propose several other research designs that could use the technique I develop to answer questions elsewhere in security studies, in comparative politics, and in American politics.

Defining Terms and Motivations

This paper focuses on one of the three steps in the complete process of turning text into structured, geolocated events. The three steps consist of:

1. Event detection, the process of recognizing that a particular kind of event is reported in text, categorizing the event, and extracting the actors involved.
2. Geoparsing, the process of recognizing place names in text and resolving them to their coordinates or to an entry in a dictionary of place names (a "gazetteer").
3. Event-location linking, connecting the event in the text to the location in the text where it is reported to have occurred.

The first process, event detection, has a long history in political science, beginning in the 1970s with manual methods (McClelland 1976; Azar 1980) and from the 1990s through today with automated methods (Schrodt, Davis, and Weddle 1994; Van Atteveldt, Kleinnijenhuis, and Ruigrok 2008; Schrodt, Beieler, and Idris 2014; Boschee et al. 2015; Beieler et al. 2016; Beieler 2016; Hanna 2017; Halterman et al. 2017).

Consider the following sentence as a running example of a sentence with actions and locations:

> After establishing a foothold in the western section of Al-Bab city, the Turkish Army and allied Syrian rebels launched an offensive on its neighbouring town of Bza'a earlier today.

An event coder, depending on the coding ontology is uses, may recognize a "taking territory"-type event and an "attack"-type event, each with their corresponding actors and targets.

Although automated methods for extracting political events are far from perfect,

Electronic copy available at: https://ssrn.com/abstract=3267476

a great deal of other work is being expending on providing general-purpose event coders and on improving their accuracy. Because my method does not depend on the specific workings of an event recognition system, only that they detect events and can identify the trigger verb of the event, I bracket further discussion of how they work.

The second step in geolocating events in text is geoparsing, which is the process of recognizing place names in text ("toponym recognition") and resolving them to their coordinates or gazetteer entry ("toponym resolution"). Some work on "geoparsing", also referred to as "georeferencing" or "toponym resolution" exists in computer science (Leidner 2008; Hill 2009; Speriosu and Baldridge 2013; D'Ignazio et al. 2014; Gritta et al. 2017; Avvenuti et al., n.d.). Some work resolving place names to coordinates also exists in political science: (Lee, Liu, and Ward 2018; Halterman 2017b; Douglass and Harkness 2018).

Returning our running example,

> "After establishing a foothold in the western section of Al-Bab city, the Turkish Army and allied Syrian rebels launched an offensive on its neighbouring town of Bza'a earlier today."

A geoparser would identify Al-Bab and Bza'a as place names and return their latitude and longitude or return ID numbers in a gazetteer. The obstacles that geoparsers must overcome include recognizing place names in the first place, including in non-standard text, and resolving ambiguous place names (e.g. "Springfield") to the correct location. While several geoparsers exist (Edinbourgh, Berico Technologies, n.d.), I use the one that I have developed (Halterman 2017b), in part because of its better accuracy, and in part to ease the integration between the geoparsing step described here and the event-location linking step described below.

The third task, and the one that this paper addresses, consists of determining for a given event where it took place, given a multiplicity of events and locations in text. For sentences that have only one event and one location, this process is straightforward. For more complex sentences, it is quite difficult. For example, in our running example, the establishment of a foothold and the beginning of an offensive occur in two different locations:

> "After establishing a foothold in the western section of Al-Bab city, the Turkish Army and allied Syrian rebels launched an offensive on its neighbouring town of Bza'a earlier today."

Unlike the geoparsing step, which resolves place names to coordinates, this event–location linking task does not necessarily require queries of an external database.

Electronic copy available at: https://ssrn.com/abstract=3267476

This task requires an understanding of the grammar of the sentence in order to link the main verb of the event with the location where it occurred. A generalized event geocoder should take in a sentence and the verb at the root of the event, and return the placename where it occurred.

Existing approaches to geocoding events in text

While no general system for geoparsing events exists, some previous work in political science and computational linguistics/natural language processing has bearing on the process of linking events and locations. I group these approaches into four types, three from political science and one consisting of computational linguistics approaches to the task.

In some limited circumstances, political scientists have used highly structured administrative data to effectively geolocate events. Text is searched for known place names and matches are used to both assign a location to events and to find that location's coordinates. For instance, Toft and Zhukov (2015) create a dataset of Chechen and Russian military operations in the Caucasus, using string matching to search for defined actors, verbs, and locations in highly constrained and structured, single-sentence descriptions of events. If all are found, an event with a location is coded from them. Similarly, Douglass and Harkness (2018) use administrative data from the British colonial government in Kenya on violent events during the Mau Mau rebellion. Events have a field to describe the location where they occurred, so no extra work to link events and locations is needed.

This first approach relies on two assumptions that are unlikely to be true in more general settings:

1. all place names used in a sentence are locations, and all locations are in the gazetteer (the dictionary assumption).
2. the only location mentioned in the text is the place where the event occurred (the one location assumption).

These assumptions are potentially plausible when studying a limited geographic area using highly structured administrative data, but are unlikely to be correct in general or in text that contains multiple events and place names.

The second approach to event geolocation in political science consists of the approach taken by production event data systems, which generally skip an event location modeling step and simply assign the locations produced by geoparsers with the events produces from the same sentence with event coders. Hanna (2017) uses CLIFF-CLAVIN (D'Ignazio et al. 2014; Berico Technologies, n.d.) on sentences

Electronic copy available at: https://ssrn.com/abstract=3267476

that have a protest event. Similarly, the Phoenix dataset arbitarily assigns the first location returned by CLIFF or Mordecai (Halterman 2017b) to the event as its location.

A third approach is one that tries to model the probability of each location in a news story being the event-occurring location, but without considering the syntactic structure of the sentence. This work is an improvement over the arbitrary linking method that most current systems use, but the current examples of this approach approach are fundamentally limited in that it does not take the event into account, and rather generates one location per sentence or document, regardless of the event of interest in the document. Lee, Liu, and Ward (2018) propose a supervised system that uses SVMs on bigrams to categorize locations in an article into four classes, determined by whether a location is the site of an event, and whether the event is the event of interest. However, because their system operates only on documents, it must be tailored to specific types of events: their model to locate protests will not perform well for locating attacks because of the differences in word usage. This requirement limits its usefulness as a general system.[3]

Imani et al. (2017) extend the concept of an article's "focus" location (D'Ignazio et al. 2014), improving techniques to find the most important or relevant city in a piece of text, and assigning any events in the text to that location. As training and test data, they use an existing dataset of hand-geolocated atrocity reports from the Political Instability Task Force, which may be closer to the administrative-type reports that Toft and Zhukov (2015) and Douglass and Harkness (2018) use. The drawback of this method is the same as in Lee, Liu, and Ward (2018), where the location is determined independently of the event, meaning that documents with multiple events and multiple locations will only return the correct location for at most one event, chosen arbitrarily. Lautenschlager, Starz, and Warfield (2017) describe a hand-created heuristic geolocation process to select a single location from several in a sentence, but do not provide details about how this process works or make their code available, it is difficult to speculate on how their system works or what its accuracy is.

For fixed event types and highly focused pieces of text, these approaches may recover the location of an event. For any diversity of event types, however, the methods will not be able to locate the events correctly because they produce one event location per document. For example, an article that describes a two separate mass killing events in rural areas and the arrival of peacekeeping forces in the capital will result in all three events being assigned the same one location.

[3]They have also not made their code available, making it impossible to compare its performance to other techniques.

Electronic copy available at: https://ssrn.com/abstract=3267476

Finally, some work in computational linguistics and natural language processing has some bearing on the event geoparsing task. The closest task in computational linguistics and natural language processing to the event–location linking task I perform is a component of the semantic role labeling FrameNet (Baker, Fillmore, and Lowe 1998) and PropBank (Palmer, Gildea, and Kingsbury 2005) tasks. Each of these tasks involve recognizing different types of "frames" or events in text and filling "slots" for each. FrameNet defines over a thousand event-like "frames", each of which has different possible slots that can be filled. Several frames have a "Location" element that can be filled. FrameNet's approach to linking verbs and locations is highly constrained: frames must first be identified, and then the location slot can be filled.

PropBank has a much more flexible approach to filling slots. Rather than each frame having very specific slots such as "patient" or "instrument", each verb has a small set of numbered arguments that loosely map to the frame elements of FrameNet. PropBank also includes a location argument, though it uses the term in a more general sense to also include non-tangible places ("keep in our thoughts). While their data is therefore not useful as evaluation data for my system, the task is similar enough that their accuracy can serve as a baseline. On their evaluation set of 95 locations, they achieve a baseline precision of 60.7 and recall of 38.9 (Palmer, Gildea, and Kingsbury 2005, 28).

Other work in computational linguistics is more tangentially related. A large literature and a set of shared tasks in computational linguistics are concerned with recognizing relationships between entities, such a employment relations between people and companies, family relations between people, or part-of-whole relations between groups and sub-groups (see Li and Ji (2014) for a baseline model). The most relevant relation type for the geoparsing task is defined in the ACE 2004 and 2005[4] corpora and is "located-in" relationship between people and locations. This relationship captures reports that a person is located in a particular place, but does not attempt to locate actions or verbs. While all current approaches to named entity recognition build on this literature, an important distinction exists between locating an entity and locating an event.

The closest work to the problem of linking events and locations is Tonon et al. (2017), which attempts to geolocate events in tweets. Their approach uses a dependency parse to label the grammatical relationships between words in a sentence, linking, for instance, a verb to the subject noun and to the object of the verb. If the root verb of the event has a location in its predicate, they assign that location

[4]https://www.ldc.upenn.edu/collaborations/past-projects/ace

Electronic copy available at: https://ssrn.com/abstract=3267476

to be the event's location. This is a good baseline approach, but will fail to find multiple locations, to select the more precise location when multiple geographic levels are provided, and to find locations in certain kinds of clauses. Because they do not create a labeled gold-standard dataset, they cannot assess the accuracy of their approach.

A supervised learning approach to linking events and locations

The major limitation of most previous work on this task is to assume that a piece of text has one distinct location where it occurred and to make the strong assumption that location and event co-occurrence in a document is sufficient information for locating the event. Because sentences can have multiple verbs, each of which may occur in different locations, an event geoparsing model must account for the word order information in the sentence and model the event location separately for each verb of interest. To illustrate this, consider the same sentence with two verbs of interest and their appropriate locations highlighted:

> He was [$_{\text{verb}}$ speaking] a day after Ankara launched an offensive in the Syrian towns of Jarablus and Kobane.[5]

> He was speaking a day after Ankara [$_{\text{verb}}$ launched] an offensive in the Syrian towns of [$_{\text{loc}}$ Jarablus] and [$_{\text{loc}}$ Kobane].

The speech event rooted by the verb "speak" in this sentence has no location: the spokesperson did not make the statement in either of the two towns, so no location is defined. The event anchored by "launch," in contrast, has two locations.

This example highlights that any attempt to find the location where a sentence or document "happened" is misguided: a sentence could be correctly located to several locations, depending on the event of interest. Previous methods that do not account for the possibility of multiple events or model event location by verb cannot serve as general systems.

The method I use to geolocate events is fundamentally a word linking task: given a piece of text and the root verb of that event, the algorithm should return the places in the text where the event occurred. This process clearly depends on word order, so the model I use for it must be able to take word order into account.

[5] Fictitious to illustrate double location

Electronic copy available at: https://ssrn.com/abstract=3267476

Data

The data I use to train and test the model consists of around 7,000 manually annotated sentences, drawn from the pro-Syrian government paper Al Masdar, Middle East Eye, ARA News, the New York Times, secondary source synopses of atrocity reports compiled by the Political Instability Task force, and Wikipedia articles. Around 4,000 were annotated by a research assistant as part of a Minerva project on territorial control in civil war, giving them greater coverage of Syria and territorial capture-type events.[6] Each sentence has a particular verb highlighted and the words describing its reported location highlighted (ranging from no words to several). These verbs were either pre-specified ("capture", "assault", "burn", etc.) or were recognized automatically in text using a dependency parser. After annotation, which occurred using a web interface[7], each sentence looks something like the following:

> He was speaking a day after Ankara [$_{\text{verb}}$ launched] an offensive in the Syrian towns of [$_{\text{location}}$ Jarablus] and [$_{\text{location}}$ Kobane].[8]

When sentences include both a specific and more general place name, the data has the more specific word highlighted and not the more general:

> Rebels [$_{\text{verb}}$ advanced] into Aleppo's [$_{\text{location}}$ Bustan al-Qasr] neighborhood.

This stands in contrast to the decisions of both Lee, Liu, and Ward (2018) and Imani et al. (2017), each of which pick cities as the lowest unit of aggregation.

Model

In contrast to previous approaches, I argue that where a sentence "took place" is only defined in relation to the event of interest contained in the sentence. To account for this, methods to geolocate sentences need to link events and place names in the sentence. This is only possible with an attempt to model the relationship between verbs and place names in the sentence, taking word order and

[6]Annotations were done by Emily Young and funded by Minerva and the U.S. Army Research Office (W911NF-13-0332).

[7]Early annotations were collected using https://github.com/Sotera/mitie-trainer, while more recent annotations were collected using Prodigy (https://prodi.gy/).

[8]Fictious to illustrate double location

Electronic copy available at: https://ssrn.com/abstract=3267476

grammatical information into account.

Baseline. One seemingly obvious approach to geolocating events would be to pick the location in the sentence in closest proximity to the event's verb. A good general rule in natural language is that words closer to each other are more likely to be related to each other, and using a named entity recognition system to find place names should increase the precision of the method. Assigning the automatically recognized location closest to a verb of interest as its location of occurrence, using spaCy's (Honnibal and Montani 2017) default English NER model to my dataset produces remarkably poor results.

table 1 Baseline proximity model accuracy

Model	Precision	Recall	F1
Word distance baseline	0.29	0.25	0.27

A precision value of 0.29 means that out of 100 locations that the baseline system flags as the location of their respective events, only 29 would indeed be the locations of occurrence. A recall of 0.25 means that out of 100 correct event locations, the system recovers only 25. The F1 score is the harmonic mean of precision and recall.

This poor performance is attributable to a few sources of error. First, the heuristic that the closest location word to a verb is where the verb took place could be incorrect. Different sentence structures, especially those with clauses, can put verbs in close proximity with different locations. The closer location word could also be a context location (e.g. the state or governorate), with the more precise and therefore correct location word occurring further away in the sentence. A second source of error comes from events that have more than one location reported. The baseline method only selects the closest, leading to poor recall in these cases. Finally, the baseline method is highly dependent on the ability of the named entity recognition (NER) system to correctly identify locations in text. Because NER systems are generally trained on American news text, they often perform badly on non-American entity names, and perform particularly badly on transliterated Arabic place names, which comprise the bulk of my labeled data.

Modeling. The very poor performance of the baseline model indicates the need for a more sophisticated modeling strategy. This modeling consists of deciding how to represent a document in the model, and how to generate an estimate of the correct location word. Ultimately, each sentence will be represented as a

Electronic copy available at: https://ssrn.com/abstract=3267476

matrix, with one row for each word, and each of these rows consisting of a vector of information about the word. A recurrent neural network, described below, then generates a probability for each word, based on its features and the other features of the sentence, that the word is the location or one of the locations where the event occurred.

Word representation. The standard way to represent a word in a bag-of-words model is as a "one-hot" vector, a vector of zeros the length of the entire vocabulary, with a single 1 in the position of the word's number in the vocabulary. This modeling approach is straightforward and easy to implement, as it just looks up the id number of each word in the vocabulary and creates a vector from it. The next step in a bag of words model would be to take the element-wise sum of the one-hot vectors to create a vector encoding the frequency of words in the document. Because the task of linking events and locations in a sentence clearly depends on word order and grammatical structure, a bag of words approach is not tenable.

Even if word order is preserved, modeling each word as a one-hot vector has a large downside. While a word-order aware model could potentially learn how to do the task given only the words in the sentence, this would require a large amount of data. The model would potentially need to learn for itself the role of grammar, which words are similar to each other, and what place names look like, among other features. We can short cut this process by representing each word not as one-hot vector encoding its identity, but as a concatenation of many features of the word, drawn from existing computational linguistics and natural language processing.

By first passing the sentence through an existing natural language processing tool, the pipeline can concatenate several other valuable pieces of information about the word and its role in the sentence. Specifically, it adds the following information:

1. word embeddings. Word embeddings are a low dimensional (usually 50-300 dimensions) representation of words, learned from their local co-occurance with other words (Mikolov et al. 2013; Levy and Goldberg 2014), on the hypothesis that "a word is characterized by the company it keeps" (Harris 1954). Word embeddings encode a great deal of semantic information about words, encoding similarity to similar words, the conjugation of verbs, and many other pieces of information based on where they occur in large quantities of unlabeled text. Word embeddings are the standard input for modern neural net-based natural language processing systems, and the other feature that most of them use.
2. dependency label. I take each word's dependency label in the sentence, as

Electronic copy available at: https://ssrn.com/abstract=3267476

predicted by spaCy. Dependency labels encode grammatical relations between words in the sentence in a directed tree format, with, for instance, a subject noun, adverb, and direct object being "dependencies" of the sentence's root verb. The direct object, in turn, may have an article and adjective as its dependencies.

3. named entity label. I use spaCy's predicted named entity label as another feature. The location tag is not highly accurate for many non-US place names, but it is still a highly informative feature. I use all labels made available through spaCy, which is trained on the OntoNotes 5 labeled English dataset (Hovy et al. 2006).[9]
4. part-of-speech tag. Cruder than dependency labels, part-of-speech taggers return a word's grammatical label, such as noun, verb, preposition, or punctuation. This feature should be much less informative than dependency information, but NLP systems achieve much better performance on part-of-speech (POS) tagging than dependency (approaching 99% accuracy for part of speech tags).
5. word distance. As discussed above the baseline section, distance between words can provide a strong signal of how related they are. I include both simple word distance as well as distance on the dependency tree.
6. verb location. Finally, each word is marked with whether it is the verb of interest or not. This feature is important for rectifying the central problem of existing event geocoders, namely that their coding of event locations is determined solely by the document and does not account for the possibility of different events having different locations in the same piece of text.

At the end, each word is represented as a vector of length 420 and each sentence is a matrix of (n_words, 420).

Sentence modeling. Neural networks are now the dominant approach to most of natural language processing. Given enough data, they offer the promise of avoiding manual feature engineering and of easily handling high-dimensional data, including text data.

There are three approaches to modeling sequence data with neural nets. The first is to concatenate the entire sequence into a single vector and use a dense, feed-forward network to take it in directly. The disadvantages of this approach are great: it requires a huge network, it has no sense of proximity in the input vector (since all inputs feed into each neuron), and no weights can be shared across

[9]The labels are described here: https://spacy.io/api/annotation#named-entities

Electronic copy available at: https://ssrn.com/abstract=3267476

words for particular features such as the weight of a part of speech tag. The second approach is to use a recurrent neural nets. Recurrent neural nets (RNN) take as inputs both a vector of data, such as a word, and a state. As words are fed through a recurrent neural net, it can learn what information from previously in the sentence and in the case of dominant "long short-term memory" (LSTM) models to learn store and what to forget from its memory cell. Conceptually, these models are the best for modeling sequence data, but are very slow to train and in practice do not live up to their theoretical benefits. The third approach, and the one that I take here, is to use a series of stacked convolutional neural networks (CNNs) to gain the speed advantages of CNNs while preserving the order benefits of recurrent neural nets. A convolutional neural net slides a set of "filters" across the input, applied the same set of weights to different regions of the input and generating a filter-specific score for each region. Each "filter" applies a weight to each input element in its sliding region and sums these products. Because each filter is applied to many regions of the input, weights can be shared across the input. In the case of images, this allows CNNs to recognize a triangle or car regardless of where in the image it occurs. In language, it allows it to learn local features of language, such as bigrams or common grammatical sequences.

I arrived at an LSTM-based neural architecture after a series of experiments on both recurrent and convolutional neural networks. For the convolutional networks, I trained a series of convolutional neural networks with residual connections. Each convolution looks at inputs at once, and slides down the sentence one at a time. By stacking them on top of each other, the elements of the final output of the fourth convolutional layer includes information from all the words in the sentence. More specifically, I use a recently developed form of convolutional networks called a deep residual network (He et al. 2016), which are now the state of the art on image recognition tasks. Residual blocks in residual neural nets perform the same convolutional step as traditional CNNs, and then element-wise add the input to the block with the block's output (see figure 2). This has several advantages. First, adding inputs helps prevent the vanishing gradient problem that deep neural nets often encounter, when the chained derivatives used to perform backpropogation to update each layer's weights result in gradients at zero (or infinity, in the case of exploding gradients). Using rectified linear unit (ReLU) as the activation function and batch normalization help address this problem, but so does adding information back in in later blocks. Second, residual layers help the model fit fast, as each block only needs to learn a small improvement to the fit. The output of previous layers passes through, a residual block, and the residual block merely adds slight improvements on top of the original input (hence residual). I used a residual CNN in my case because they empirically outperformed a model of similar depth and structure without residual blocks, and theoretically

Electronic copy available at: https://ssrn.com/abstract=3267476

because they allow me to train a deeper network with lower demands on my limited pool of input data. After training and evaluating several dozen models, the best performing CNN model used 7 recurrent layers with 64 hidden nodes in each, followed by two dense layers with 512 nodes each with a dropout of 0.4 and ReLu activation.

The second class of models I fit were recurrent neural networks (RNNs), specifically LSTM networks (Hochreiter and Schmidhuber 1997), which explicitly model the sequential nature of their input data. LSTMs store an internal state at each step of the input data in the form of a hidden vector. In contrast to vanilla RNNs, LSTMs can learn when to add information from their current input step to the hidden state and when to "forget" information from the hidden state. In theory, this allows LSTMs to learn much longer relationships than they would otherwise be able to. Bidirectional LSTMs are the standard extension to LSTMs when the model has access to the "future", and compute two state vectors for each input step: one from the left and one from the right. These two vectors are concatenated and used as input to the rest of the model. The best LSTM network I trained used a bidirectional LSTM with a hidden size of 128 and 0.2 recurrent droput, followed by a dense layer of 128 with ReLu and 0.5 dropout, and a final binary output node for each time step.

Evaluation. I evaluated each candidate model on an evaluation set. In each experiment, I recorded the loss, accuracy, and precision and recall of each model on a held-out evaluation set, picking the best model of these. The best CNN and LSTM models yielded the following precision and recall on a held-out test set:

table 2 Model-based approaches compared to closest result in the literature and word distance baseline.

Model	Precision	Recall	F1
Word distance baseline	0.29	0.25	0.27
~~FrameNet~~ (Palmer, Gildea, and Kingsbury 2005)	0.61	0.39	0.47
CNN	0.70	0.54	0.61
LSTM	0.82	0.82	0.82

All three non-baseline models clearly dominate the word distance baseline model. The FrameNet values are reported for the ArgM-LOC label in Palmer, Gildea, and Kingsbury (2005). They specify the location task in a slightly different form, but their performance is a relatively close analog in the published literature. The LSTM model performs much better than the CNN model, even after extensive tuning for the CNN model. Inspection of the CNN model's output indicates

Electronic copy available at: https://ssrn.com/abstract=3267476

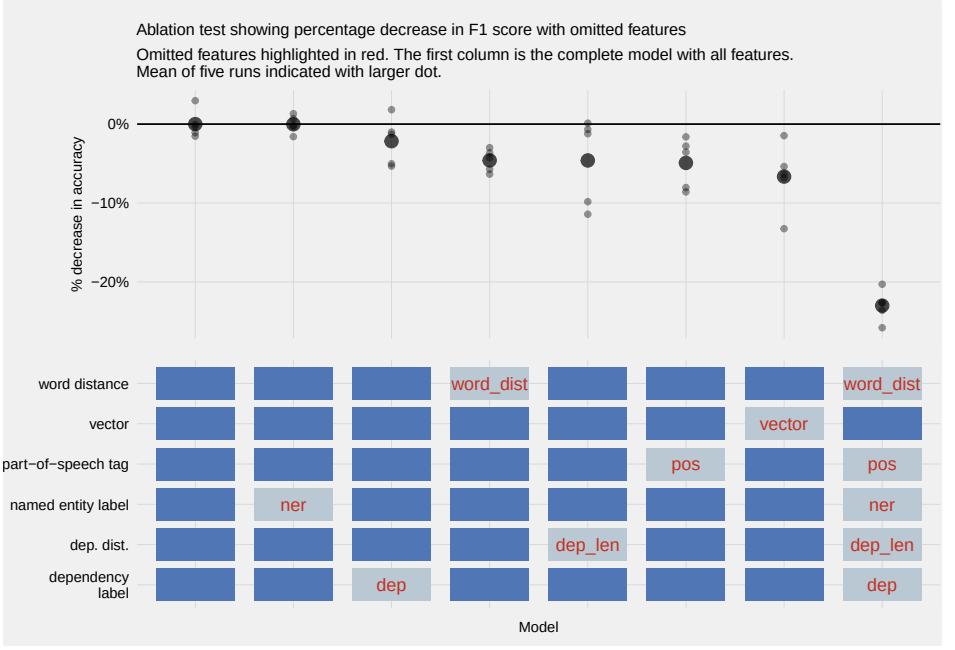

Figure 1. "Ablation test" showing decrease in F1 score with omitted features. Full model includes dependency labels (deps), pre-trained GloVe embeddings (vector), part-of-speech tags (pos), named entity labels (ner), the (signed) distance from the word to the verb (word_dist), and the length of the dependency path from the word to the verb (dep_len). All conditions used the same neural net model, with the best performing model on a validation set applied to held out test set.

that the model seems to not learn long-distance relationships as well as needed, and failed to appropriately change probability weights when the verb of interest changed. The LSTM model, in contrast, performs very well and is very sensitive to changes in the input verb.

Because the weights neural networks place on their inputs are not highly interpretable and the importance of features can change in interaction with others, it can be difficult to understand the important of different features in the model. One approach is to conduct an "ablation" test, iteratively removing features from the model, retraining it and comparing its performance. Figure 1 shows the results of this process on the best performing LSTM model, revealing that some features are more important than others across several random partitions and retrainings of the model.

Electronic copy available at: https://ssrn.com/abstract=3267476

The ablation test reveals several interesting findings. First, the variability in feature importance across different train-test splits of the data prevents too-strong claims. With that in mind, the named entity label returned by spaCy would seem to be a useful feature in a task that requires picking one of potentially several place names. In fact, removing it leaves the accuracy unchanged, perhaps because the labeled data skews toward Arabic place names, which spaCy struggles to recover. The two distance features, one encoding distance from each word to the verb of interest and the other encoding the length of the shortest dependency path between them, both seem marginally helpful. Surprisingly, the part-of-speech feature is more useful than the dependency label. This may be because the tree structure of the dependency parse is not being incorporated, only its labels are. Finally, the pretrained GloVe embedding feature is helpful (second to the right column), but it is by no means sufficient on its own (rightmost column). While some of the literature on neural networks for NLP simply starts from pretrained word or character embeddings and learns useful representations from those, these results indicate that wider feature inclusion is very helpful for the model's accuracy.

A less formal evaluation comes from inspecting the LSTM model's output on several sentences. Table ?? visually displays the output of the model on the example sentence used above.

The model easily flags that "establishing a foothold" occured at "Al-Bab", and not at "Bza'a", though the probability of "Bza'a" being the location is higher than for other words in the sentence.[10] In fact, there are two verbs (or events) in this sentence: the establishment of foothold and the launching of an offensive. Running the same sentence through the model with "launch" flagged as the verb of interest dramatically changes the output of the model:

[10]Longer training for more epochs causes the model to begin to flag all place names as correct locations. Under-training the model results in fewer extreme probabilities (0 and 1).

Electronic copy available at: https://ssrn.com/abstract=3267476

table 3 Output of the method on an example sentence

Predicted	Prob	Word
-	0.00	After
VERB	0.00	establishing
-	0.00	a
-	0.00	foothold
-	0.00	in
-	0.00	the
-	0.00	western
-	0.00	section
-	0.00	of
True	1.00	Al
True	1.00	-
True	0.98	Bab
-	0.00	city
-	0.00	,
-	0.00	the
-	0.00	Turkish
-	0.00	Army
-	0.00	and
-	0.00	allied
-	0.00	Syrian
-	0.00	rebels
-	0.00	launched
-	0.00	an
-	0.00	offensive
-	0.00	on
-	0.00	its
-	0.00	neighbouring
-	0.00	town
-	0.00	of
-	0.22	Bza'a
-	0.00	earlier
-	0.00	today
-	0.00	.

Electronic copy available at: https://ssrn.com/abstract=3267476

table 4 The same sentence with ”launch” flagged

Predicted	Prob	Word
-	0.00	After
-	0.00	establishing
-	0.00	a
-	0.00	foothold
-	0.00	in
-	0.00	the
-	0.00	western
-	0.00	section
-	0.00	of
-	0.00	Al
-	0.00	-
-	0.00	Bab
-	0.00	city
-	0.00	,
-	0.00	the
-	0.00	Turkish
-	0.00	Army
-	0.00	and
-	0.00	allied
-	0.00	Syrian
-	0.00	rebels
VERB	0.00	launched
-	0.00	an
-	0.00	offensive
-	0.00	on
-	0.00	its
-	0.00	neighbouring
-	0.00	town
-	0.00	of
True	1.00	Bza’a
-	0.02	earlier
-	0.00	today
-	0.00	.

Electronic copy available at: https://ssrn.com/abstract=3267476

Much of the error results from the very specific labels of the training and evaluation data. When a location in the sentence is a more general area encompassing a specific location ("the Bustan al-Qasr neighborhood of Aleppo") only the more specific term is labeled as correct. The model struggles to differentiate this pattern of multiple locations from sentences that contain two locations for an event ("the strikes targeted Aleppo and Idlib provinces"). This confusion is probably not resolvable without access to information in a gazetteer. Performing the geoparsing step first, and then incorporating that inofrmation into the event linking step could improve accuracy here.

End-to-End example: Geolocating offensives in Syria

To demonstrate the usefulness of this approach, I use it to create a dataset of Syrian military offensives in 2016 by automatically coding military offensive events from text and geolocating them.

Event coding. I collected 15,229 news stories on Syria covering 2016 from four sources: Al-Masdar news, Middle East Eye, Ara News, and news put out by the opposition National Coalition. Although event detection is not the main subject of this work, I will describe the process I used to generate the events. To recognize the events themselves in the text, I performed a dependency parse of all the documents in the corpus.[11] For each sentence, I checked whether a verb in the sentence, after lemmatization, was in the following list: "launch", "begin", "initiate", "start", "continue", "join", "renew". If it was, I then checked whether the verb had as a direct object one of the following terms: "attack", "offensive", "operation", "assault", "counter-attack", "counter-offensive", "counter-assault", "advance". Using a grammar-based approach avoids the need for complicated regular expressions in order to match more complicated phrases such as "launched an imperative and long-anticipated offensive". Using the dependency parse and this process of checking verbs and their direct objects is very easy to implement for a very defined type of event and produces very few false positives.[12]

After recognizing an event in the text, I then use my event geoparsing method to

[11] A dependency parse encodes the grammatical relations between words. I used spaCy (https://spacy.io) to parse the documents.

[12] The main problem now is in distinguishing between ongoing attacks and past attacks being re-described. I could resolve this problem with relatively little work.

Electronic copy available at: https://ssrn.com/abstract=3267476

find the location in the text linked to the event's verb. In order to produce final usable event data, I also perform the final step of resolving the event location to its geographical coordinates. To do so I use the Mordecai text geoparser (Halterman 2017b), which uses a neural net trained on several thousand gold-standard resolved place names to infer the country of a location mention, perform a fuzzy-string search over the Geonames gazetteer (Wick and Boutreux 2011), and to select the best location among the locations returned from the search.[13]

Applying the event extraction, event geoparsing, and placename resolution steps to the following sentence produces the structured representation below:

> On Sunday, rebel forces had otherwise launched two consecutive assaults on nearby Qabasin, a highly embattled town under ISIS control.

returns an event with 'rebel forces' as the actor and 'Qabbāsīn' (36.30774', 38.15598) as the location where the "launch […] assault" event took place.

The method works even for locations at finer resolutions than cities, finding, for instance, that the Kindi Hospital in Aleppo is at 36.26718, 37.1828. Some events are not geolocated because some locations are not in the gazetteer, such as the 'Al-Sha'ar Gas Fields'. When the event location step is correct, the geoparsing step can still create errors by selecting the wrong location from the gazetteer of place names. The following sentence should be resolved to Aleppo the city, but Mordecai resolved it to Aleppo Governorate, which is a clear error:

> "Jabhat Al-Nusra (Syrian Al-Qaeda group) – backed by Jaysh Al-Mujahiddeen, Harakat Nouriddeen Al-Zinki, and the Free Syrian Army (FSA) – launched a new offensive in the northwestern sector of Aleppo City on Monday morning, targeting the government-held districts of Al-Khalidiyah and Al-Zahra."

Figures 3 and 4 show the geographic and temporal distribution of new offensives. Both show that at a high level of aggregation, no major missingness is occurring.

When combined with geolocated data on civilian deaths in Syria (Halterman 2017a), the geolocated offensives allow us to determine that 7.4% of civilian deaths in Syria occurred within one day and 1 kilometer of an announced military operation.

Many existing theories of violence against civilians in civil war discount the role that conventional fighting between two sides can result in civilian casualties (Hal-

[13] A software implementation is available: "Mordecai: Full text geoparsing as a Python library" https://github.com/openeventdata/mordecai

Electronic copy available at: https://ssrn.com/abstract=3267476

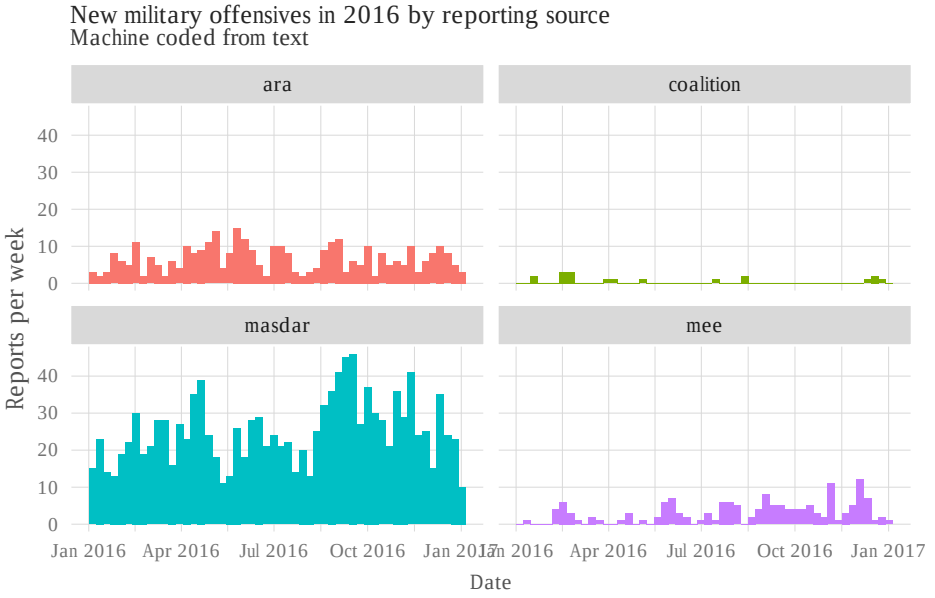

Figure 2. Number of reported offensives per week in Syria in 2016.

Electronic copy available at: https://ssrn.com/abstract=3267476

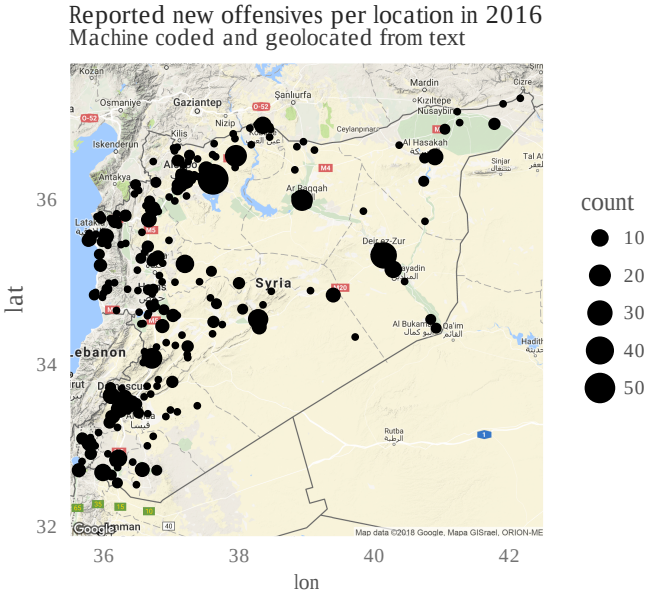

Figure 3. Locations of reported offensives in Syria in 2016

Electronic copy available at: https://ssrn.com/abstract=3267476

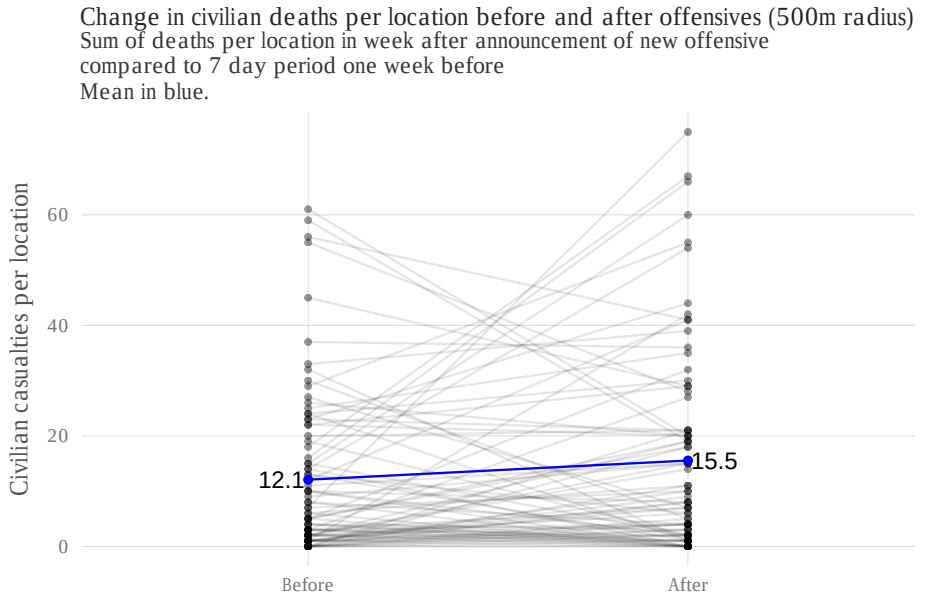

Figure 4. Change in civilian deaths per location per week before and after new offensives within 500 meters

Electronic copy available at: https://ssrn.com/abstract=3267476

terman 2017a). The ability to create a dataset of when and where conventional fighting is occurring paves the way for better understanding of the patterns of violence against civilians in civil wars.

Future work and Conclusion

This method is much more general than a technique for geolocating offensives in civil war. Any text that includes descriptions of events and locations could have this process performed on it, and any question that involves measuring subnational variation could potentially be addressed using the data this method produces. To demonstrate the broader applicability of the method, I sketch out several possible research designs below.

The order of process could also be reversed: rather than starting with an event and finding its locations, all locations can be linked back to events. Then, in a more exploratory manner, potentially with an unsupervised topic model, the activities going on in a particular area could be defined without the need to establish a priori event definitions.

Pre-war protests and government retaliation. Another application for geolocated event data is in the study of violence in civil war. Balcells (2010) argues that indiscriminate government violence against civilians in civil war is an attempt to eliminate areas of pre-war political opposition to the government. Studying the Spanish Civil War, she can draw on pre-war voting returns to measure opposition to the government. This measure, however, is not widely available in other civil war contexts. An alternative measure of anti-regime political mobilization could be pre-war protest activities, which convey much of the same information. The techniques I employ above could be used to create a finely geolocated measure of protest activity, which could then be used to extend Balcell's theory beyond Spain and other cases where voting data exists. Creating a fine-grained protest datasets by hand is a common activity for scholars studying the effects of protest on political opinions (Wallace, Zepeda-Millán, and Jones-Correa 2014), the effect of local political opportunities on protest activity in Russia (Lankina 2015), the role of regime type in moderating anti-FDI protests (Robertson and Teitelbaum 2011), and the involvement of the Chinese government in alternatively encouraging or preventing anti-Japanese protests. Many of these tasks could easily be accomplished with a computer.

Electronic copy available at: https://ssrn.com/abstract=3267476

Voter turnout. Fine-grained geolocated data from text can also be used to address one of the major questions in American politics, namely the causes of voter turnout and what campaigns can do to affect it (Hillygus 2005; Gerber, Green, and Larimer 2008; Geys 2006). Politicians and campaigns devote large portions of their time and money budgets to local campaign stops, but the effects of these stops are not well understood. Previous work has studied the effects in the 2000 election of presidential and vice-presidential candidate appearances in a state on state-level opinion polls (Hill, Rodriquez, and Wooden 2010), but data availability prevented them from studying the local effects of an appearance or from studying non-presidential appearances. Automated methods could create data with the town or city location of appearances from news text, giving researchers the ability to compare locations with and without campaign stops to measure the effect of a campaign stop on turnout and the geographic attenuation of the effect. geographic attenuation

Military diplomacy. A major debate in security studies and foreign policy is over non-military uses of armed forces in promoting national interests, including through US Navy port calls and "show the flag" visits (Spindel) or through US joint military exercises and training. Militaries encourage local press coverage of these activities, but do not report (and potentially do not even collect) comprehensive data on all exercises or visits. For researchers interested in the effects of these visits and joint exercises on domestic political opinion or cooperation with the visiting country, a comprehensive dataset of the events and their locations is an important first step.

Each of these applications have endogeneity concerns that need to be addressed, but having useable data on the variables in each of these cases is a prerequisite for quantitative study. Still, these potential applications demonstrate the usefulness of word order-aware text processing methods to produce rich new datasets that were previously infeasible to create.

Conclusion

This paper introduces a state-of-the-art technique for linking events and locations in text. It proposes a new conceptualization of this task, focusing more on broad applicability than previous approaches in natural language processing, but more carefully accounting for grammar and the potential multiplicity of events than previous work in political science. I created a labeled corpus of events and their locations, and use it to train an LSTM model. This model achieves an F1 score of 0.82, making it accurate enough for researchers to begin to use.

Electronic copy available at: https://ssrn.com/abstract=3267476

More broadly, this work builds on a growing body of research in political science that attempts to extract information from text, rather than summarizing or categorizing text. A wealth of techniques, especially topic models, exist for researchers to make inference about the discursive or thematic content of text. Text also holds a great deal of factual information, though, and new techniques are needed to allow researchers to extract information from text. The technique introduced here will improve researchers' ability to incorporate information extracted from text into research studies that rely on geographically fine-grained data.

Electronic copy available at: https://ssrn.com/abstract=3267476

References

Avvenuti, Marco, Stefano Cresci, Leonardo Nizzoli, and Maurizio Tesconi. n.d. "GSP (Geo-Semantic-Parsing): Geoparsing and Geotagging with Machine Learning on Top of Linked Data."

Azar, Edward E. 1980. "The Conflict and Peace Data Bank (COPDAB) Project." Journal of Conflict Resolution 24 (1): 143–52.

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, 86–90. Association for Computational Linguistics.

Balcells, Laia. 2010. "Rivalry and Revenge: Violence Against Civilians in Conventional Civil Wars." International Studies Quarterly 54 (2): 291–313.

———. 2017. Rivalry and Revenge: The Politics of Violence During Civil War. Cambridge University Press.

Balcells, Laia, and Stathis N Kalyvas. 2014. "Does Warfare Matter? Severity, Duration, and Outcomes of Civil Wars." Journal of Conflict Resolution 58 (8): 1390–1418.

Beieler, John. 2016. "Generating Politically-Relevant Event Data." CoRR abs/1609.06239. http://arxiv.org/abs/1609.06239.

Beieler, John, Patrick T Brandt, Andrew Halterman, Erin Simpson, and Philip A Schrodt. 2016. "Generating Political Event Data in Near Real Time: Opportunities and Challenges." In Data Analytics in Social Science, Government, and Industry, edited by R. Michael Alvarez. Cambridge University Press.

Berico Technologies. n.d. "CLAVIN: Cartographic Location and Vicinity Indexer."

Boschee, Elizabeth, Jennifer Lautenschlager, Sean O'Brien, Stephen M Shellman, James Starz, and Michael D Ward. 2015. "ICEWS Coded Event Data." In Http://Dx.doi.org/10.7910/Dvn/28075. Harvard Dataverse, V9.

Douglass, Rex W, and Kristen A Harkness. 2018. "Measuring the Landscape of Civil War: Evaluating Geographic Coding Decisions with Historic Data from the Mau Mau Rebellion." Journal of Peace Research.

D'Ignazio, Catherine, Rahul Bhargava, Ethan Zuckerman, and Luisa Beck. 2014. "CLIFF-CLAVIN: Determining Geographic Focus for News." NewsKDD: Data

Electronic copy available at: https://ssrn.com/abstract=3267476

Science for News Publishing, at KDD 2014.

Gerber, Alan S, Donald P Green, and Christopher W Larimer. 2008. "Social Pressure and Voter Turnout: Evidence from a Large-Scale Field Experiment." American Political Science Review 102 (1). Cambridge University Press: 33–48.

Geys, Benny. 2006. "Explaining Voter Turnout: A Review of Aggregate-Level Research." Electoral Studies 25 (4). Elsevier: 637–63.

Gritta, Milan, Mohammad Taher Pilehvar, Nut Limsopatham, and Nigel Collier. 2017. "What's Missing in Geographical Parsing?" Language Resources and Evaluation. Springer, 1–21.

Halterman, Andrew. 2017a. "Violence Against Civilians in Syria's Civil War." MIT Political Science Department Research Paper.

———. 2017b. "Mordecai: Full Text Geoparsing and Event Geocoding." The Journal of Open Source Software 2 (9). doi:10.21105/joss.00091.

Halterman, Andrew, Jill Irvine, Manar Landis, Phanindra Jalla, Yan Liang, Christan Grant, and Mohiuddin Solaimani. 2017. "Adaptive Scalable Pipelines for Political Event Data Generation." In IEEE International Conference on Big Data, 2879–83. IEEE.

Hanna, Alex. 2017. "MPEDS: Automating the Generation of Protest Event Data." SocArXiv.

Harris, Zellig S. 1954. "Distributional Structure." Word 10 (2-3). Taylor & Francis: 146–62.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. "Deep Residual Learning for Image Recognition." In Proceedings of the Ieee Conference on Computer Vision and Pattern Recognition, 770–78.

Hill, Jeffrey S, Elaine Rodriquez, and Amanda E Wooden. 2010. "Stump Speeches and Road Trips: The Impact of State Campaign Appearances in Presidential Elections." PS: Political Science & Politics 43 (2). Cambridge University Press: 243–54.

Hill, Linda L. 2009. Georeferencing: The Geographic Associations of Information. MIT Press.

Hillygus, D Sunshine. 2005. "Campaign Effects and the Dynamics of Turnout Intention in Election 2000." Journal of Politics 67 (1). Wiley Online Library: 50–68.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory."

Electronic copy available at: https://ssrn.com/abstract=3267476

Neural Computation 9 (8). MIT Press: 1735–80.

Honnibal, Matthew, and Ines Montani. 2017. "SpaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing." To Appear.

Hovy, Eduard, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. "OntoNotes: The 90% Solution." In Proceedings of the Human Language Technology Conference of the Naacl, Companion Volume: Short Papers, 57–60. Association for Computational Linguistics.

Imani, Maryam Bahojb, Swarup Chandra, Samuel Ma, Latifur Khan, and Bhavani Thuraisingham. 2017. "Focus Location Extraction from Political News Reports with Bias Correction." In Big Data (Big Data), 2017 Ieee International Conference on, 1956–64. IEEE.

Kalyvas, Stathis N. 2006. The Logic of Violence in Civil War. Cambridge University Press.

Lankina, Tomila. 2015. "The Dynamics of Regional and National Contentious Politics in Russia: Evidence from a New Dataset." Problems of Post-Communism 62 (1). Taylor & Francis: 26–44.

Lautenschlager, Jennifer, James Starz, and Ian Warfield. 2017. "A Statistical Approach to the Subnational Geolocation of Event Data." In Advances in Cross-Cultural Decision Making, 333–43. Springer.

Lee, Sophie J, Howard Liu, and Michael D Ward. 2018. "Lost in Space: Geolocation in Event Data." Political Science Research and Methods, 1–18. doi:arXiv preprint arXiv:1611.04837.

Leidner, Jochen L. 2008. Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding of Place Names. Universal-Publishers.

Levy, Omer, and Yoav Goldberg. 2014. "Neural Word Embedding as Implicit Matrix Factorization." In Advances in Neural Information Processing Systems, 2177–85.

Li, Qi, and Heng Ji. 2014. "Incremental Joint Extraction of Entity Mentions and Relations." In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 1:402–12.

McClelland, Charles A. 1976. "World Event/Interaction Survey Codebook." ICPSR Ann Arbor, MI.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. "Distributed Representations of Words and Phrases and Their Compositionality."

Electronic copy available at: https://ssrn.com/abstract=3267476

In Advances in Neural Information Processing Systems, 3111–9.

Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. "The Proposition Bank: An Annotated Corpus of Semantic Roles." Computational Linguistics 31 (1). MIT Press: 71–106.

Robertson, Graeme B, and Emmanuel Teitelbaum. 2011. "Foreign Direct Investment, Regime Type, and Labor Protest in Developing Countries." American Journal of Political Science 55 (3). Wiley Online Library: 665–77.

Schrodt, Philip A, John Beieler, and Muhammed Idris. 2014. "Three's a Charm?: Open Event Data Coding with EL:DIABLO, PETRARCH, and the Open Event Data Alliance."

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). Sage Publications Sage CA: Thousand Oaks, CA: 561–87.

Speriosu, Michael, and Jason Baldridge. 2013. "Text-Driven Toponym Resolution Using Indirect Supervision." In ACL (1), 1466–76.

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): 222–38.

Tonon, Alberto, Philippe Cudré-Mauroux, Albert Blarer, Vincent Lenders, and Boris Motik. 2017. "ArmaTweet: Detecting Events by Semantic Tweet Analysis." In European Semantic Web Conference, 138–53. Springer.

Van Atteveldt, Wouter, Jan Kleinnijenhuis, and Nel Ruigrok. 2008. "Parsing, Semantic Networks, and Political Authority Using Syntactic Analysis to Extract Semantic Relations from Dutch Newspaper Articles." Political Analysis 16 (4). Oxford University Press: 428–46.

Wallace, Sophia J, Chris Zepeda-Millán, and Michael Jones-Correa. 2014. "Spatial and Temporal Proximity: Examining the Effects of Protests on Political Attitudes." American Journal of Political Science 58 (2). Wiley Online Library: 433–48.

Weinstein, Jeremy M. 2006. Inside Rebellion: The Politics of Insurgent Violence. Cambridge University Press.

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

Electronic copy available at: https://ssrn.com/abstract=3267476