[DRAFT] Learning Events From Text

Andy Halterman

13 July 2019

Abstract

Measuring what political actors do in the world at the core of empirical social science, and researchers expend enormous effort to compile data on events and behavior from text. I introduce a method for inductively and automatically learning political events from text. Qualitative coding of events is slow, expensive, and limits researchers' abilities to explore large collections of text, while existing quantitative methods are brittle, expensive to create, and not suited to inductively learning event categories from text. I decompose the problem of learning events into two components. The first is to take in text and identify the key elements of an "event": who is doing what to whom, where and when, as reported by whom? Each of these is a "slot" to be filled. I introduce a method for recognizing the span of text associated with each slot that combines information from the dependency parse and a machine learning classifier that distinguishes between words that occupy the same grammatical role in a sentence but fill different slots (e.g. "fired missiles" vs. "fired Tillerson"). The second task is to aggregate these extracted spans in a way that is useful for researchers. I introduce a new short text clustering algorithm that draws on prior information in the form of word embeddings to learn different event types from text. These techniques are not domain specific, so researchers can apply it to their own questions in the same way they would use topic models. I apply the method to an ongoing debate in international politics about the level of respect for human rights. I extract and cluster over one million events reported in State Department reports and find that although the number of reported violations is increasing, the types of actions and the specificity of each have changed over time, suggesting a changing standard of reporting.

Much of the data used by social scientists to understand the world consists of descriptions of events produced by hand from text (Table 1). Text sources, including encyclopedias, newspaper reports, NGO project documentation, and civil society analyses are the raw material used to build datasets with information on political events, which report "who did what to whom, when and where?" Structured data on political events, consisting of information on actors and behavior in the world, is a crucial form of data in political science, especially as the field progresses further toward micro-level study of social phenomena. Yet our existing and growing set of automated text analysis methods are not built primarily for summarizing documents, not for extracting information from them. Existing techniques for extracting data on "who did what to whom" from text in political science and computer science have several shortcomings. They often require enormous up-front effort to customize systems to new event types, they are trained to produce events of a type that are irrelevant to political science, and they have little to no ability to inductively learn event types from text. As in the rest of science, the availability of new data is often the precipitating cause of new research and improved understanding. Automating some production of structured data from text would allow more project-specific creation of data, leading to better

measurement strategies that use better data that is customized to the question at hand, and ultimately, improved understanding of the world.

Dataset	Citation	Citation Count	Text Sources
MIDS	Jones, Bremer, and Singer (1996)	2171	diplomatic sources, histories, newspapers
CIRI human rights	Cingranelli and Richards (2004)	55 (?)	State Dept. reports
GTD	LaFree and Dugan (2007)	455	newswire, newspaper, gov. documents
Archigos	Goemans, Gleditsch, and Chiozza (2009)	670	Encyclopedias, newspapers
ACLED	Raleigh et al. (2010)	845	news text and humanitarian reporting
coups	Powell and Thyne (2011)	317	NYT and other text sources
SCAD	Salehyan et al. (2012)	270	AP and AFP
SIPRI arms transfers			commercial publications, newspapers, gov. publications
UCDP intrastate conflict	Sundberg, Eck, and Kreutz (2012)	174	newspapers
SPEED "civil strife"	Nardulli, Althaus, and Hayes (2015)	26	NYT, BBC Monitoring, FBIS
regime type	Geddes, Wright, and Frantz (2014)	542	News reports, published literature

Table 1: Many structured datasets are derived from text sources. Producing and updating them is often a multi-year, multi-annotator undertaking.

This project introduces two new techniques in automated text analysis that together make it possible for researchers to extract event information and then usefully aggregate similar events from text. My method requires two steps: a "slot filling" step that extracts the words from a sentence that correspond to the actors, actions, and other information around an event, and a second "aggregation" step that groups the extracted phrases into categories for analysis. Previous methods have required large up-front human labor to build dictionary-based event recognition models and required rigid, pre-specified ontologies of actors and events. My new techniques allow researchers to learn events inductively from text using a single model that generalizes across domains without the need for retraining.

The first technique I introduce is a method for recognizing the spans of text associated with each of the "slots" of a political event. I propose a new set of standard slots that generalize across event types, consisting of actors who do an action, the action itself, the political entity receiving the action, and the means or instrument involved in the action, along with slots for the reported cause or reason for the event, any reporter or source attribution

in the text, and the date and location of the event. To fill these slots in practice, I introduce a technique that combines a rule-based system that uses the grammatical information of the sentence to identify spans of text that potentially correspond to each slot, and machine learning models trained on labeled spans to determine the correct event slot for each span of words.

The second technique performs the aggregation step, learning which actions belong together using a new text clustering algorithm. Rather than relying on dictionaries or supervised models to categorize events, it instead learns event types inductively. Because extracted action spans are often quite short, traditional topic models do not perform well. Instead, I uses pretrained embeddings to provide prior knowledge on word similarity and an iterative model to cluster very short phrases into useable classes of events. I show that this model outperforms standard topic modeling approaches in a simulation setup, and produces qualitatively good results on real text.

Finally, I demonstrate the utility of these new techniques to answer substantive questions in political science by returning to the ongoing debate on whether respect for human rights has improved over time. I produce new, disaggregated data on the specific acts of human rights abuses reported by the State Department in their monitoring documents over time. I offer clear evidence that the contents of reporting are changing over time, and suggestive evidence that the threshold for inclusion are changing as well. This application demonstrates the importance of creating new, tailored data answer open questions in political science.

Producing events from text

Political "events" are structured representations of political behavior, consisting of information on actions, the involved actors, and information the manner, place, and time of the actions. Which actors or actions are "political" or relevant will depend on the specific research question at hand.

Extracting events from text as a task can be conceptually decomposed into two sequential steps: slot filling and aggregation. Slot filling involves identifying the shorter pieces of text that correspond to various attributes of the event, such as its location, the actor performing the action, or any "instrument" used in performing the event. Aggregation is a second step of putting similar entities together in a category (for instance, "fight" and "attack" might belong together, while "provide aid" and "deliver food" belong together in a separate category). Aggregation can be done in a supervised way, with spans assigned to clusters using a trained machine learning model or, more commonly, with hand-created dictionaries. It can also be done in an unsupervised way, with

categories learned inductively from the collection of spans. How spans are resolved to categories will depend on the specific research question being asked.

Slot filling

Slot filling is the task of determining which words in a sentence correspond to different "slots" in an event, including the actor doing the event, the verb and predicate information describing the event, the recipient of the action, and potentially other information such as the date of the event and the source reporting the event. I review the existing approaches to slot filling in computer science and political science and argue that the for political events, the major outstanding obstacle is what I term the "direct object" problem, of determining when objects of verbs are "instruments" of the action, and when they are "recipients" of the action. Instruments are objects used in the commission of the action, while recipients are political entities that receive the action. Recipients and instruments cannot be distinguished on the basis of grammar alone. Instead, identifying which nouns are recipients and which nouns are instruments requires substantive knowledge to distinguish them. This requirement for substantive knowledge explains why the Computer science literature has not yet produced a useful political event extraction system.

Previous work on slot filling

A wide body of literature on slot filling and "semantic role labeling" exists in computer science and natural language processing, attempting to create systems that can faithfully reflect the tremendous variety of human language and human behavior. Early semantic role labeling approaches have highly variable "slots" that differ by the recognized event type. FrameNet (Baker, Fillmore, and Lowe 1998), for instance, specifies around 1,000 linguistic "frames." Many of these slots are specific to the event type: a "cook food" event, for instance, might have a slot for "source of heat". Many event types, however, have slots that are roughly comparable: a "crime" event's "victim" slot is roughly comparable to a "hire" event's "worker". Slots can only be filled once the type of event has been recognized, making automated approaches to slot filling difficult. For political scientists, some of these frames involve potentially political actions such as a "revenge" frame, specifying the injured party, the victim, and the manner of revenge. Other frames are less interesting to political scientists: a "clothing" frame includes roles for garment, material, color descriptors, and wearer. This approach suffers from several drawbacks for applied information extraction work. First, these themes must be laboriously constructed by expert linguists,

and their great level of specificity is aimed more at linguistic correctness than at practical usefulness (for example, great care is taken to distinguish bank deposits from alluvial silt deposits, or a "killer" role in murder from the "perpetrator" role in a kidnapping). Second, as with all hand constructed dictionary methods, it faces problems of low recall (Pavlick et al. 2015). Finally, the specificity of the slots makes the system difficult to train. A "victim" of a crime and a "beneficiary" of a gift both receive an action in some sense but FrameNet treats them as completely different entities. Building on theoretical insights by Dowty (1991) on "proto-agents" and "proto-patients", Palmer, Gildea, and Kingsbury (2005) developed a much more general approach to the task, replacing specific frame elements with more general, numbered arguments. The role of each numbered argument still varies, however, by the nature of the verb, making it nontrivial to extract all agents or patients.

A second strand of literature in computer science and natural language processing is on "open information extraction" (open IE) which flips the priority of the steps. Open IE systems have the same slots for all event types, but make little attempt to distinguish between different kinds of events or states of being. A canonical open IE example shows how the sentence "President Obama was born in Hawaii" would produce two pieces of information: [(Obama, is, president), (Obama, born in, Hawaii)]. The primary application of open IE has been in knowledge base creation. These information triplets are often not events at all, and are often difficult to resolve to defined event types.

The most promising existing approach from natural language processing to the span labeling task is PredPatt (Rudinger and Van Durme 2014; White et al. 2016), which uses deterministic rules on a universal dependency parse to label the arguments of an event. A rule-based system eliminates the need for training data. PredPatt will not work for the political event extraction task task without modification, however. First, important information about the role of actors is lost in prepositional phrases. As they put it,

> " 'Mary stuffed envelopes with coupons' and 'Mary stuffed envelopes with John' have identical dependency structures, yet 'coupons' and 'John' are (hopefully for John) taking on different semantic roles" (Rudinger and Van Durme 2014, 57).

Distinguishing between political actors and other objects is crucial for political science applications. PredPatt also cannot overcome the PropBank problem of not producing labels (e.g. agent, patient) instead of PropBank's more generic numbered arguments.

The tradition in political science, in contrast, has been to limit the number of slots and use the same set across all event types. In dominant automated approaches, events usually have three slots: a "source" actor, an action,

and a "target" actor.[1] Early systems filled slots entirely using dictionary methods (spans of text matching a list of actions were assigned to action slots) (e.g. Schrodt, Davis, and Weddle 1994; Schrodt 2009; Boschee et al. 2015). Later systems (e.g. Norris, Schrodt, and Beieler 2017) use grammatical information about the sentence in conjunction with dictionary information to perform the slot filling task. Dictionary-based methods require enormous up-front investment, have very low recall between 5% and 35% (Makarov 2018; Althaus, Peyton, and Shalmon 2018), and are difficult to extend to new event types. Moreover, systems that depend on dictionaries cannot be used to learn new event types inductively from text. An approach at the intersection of the two has been to directly classify sentences into one of four broad categories of events (cooperative and conflictual, verbal and material), but without an attempt to extract spans (Beieler 2016). This approach is only suitable though for extremely coarse event types.

A promising hybrid approach comes from O'Connor, Stewart, and Smith (2013), which uses dictionaries to identify actors (here, countries) and grammatical information from the dependency tree to fill the "action" slot linking the two actors. A modified topic model that accounts for temporal dependency in dyadic relationships learns events inductively. This approach still depends on pre-built dictionaries, however.

Similarly, Van Atteveldt et al. (2017) use handwritten rules and the produced dependency parse of the sentence to segment the sentence into actor, predicate, and "source." This approach is very similar to that of PredPatt (Rudinger and Van Durme 2014; White et al. 2016). While innovative in its use of dependency parses to model actions with sentences, this model has a major limitation. It combines actions and the recipients of actions together into a single predicate span. Doing so makes it impossible to distinguish which words are political entities receiving actions. They use as an example sentence, "Hospital officials in Gaza said that 390 people were killed by Israeli fighter planes." Their method returns "390 people [were] killed" as a single predicate span, rather than separating out "killed" as an action and "390 people" as a target (in my terminology, recipient) of that action.

An ontology of event slots

I propose an ontology of event slots that builds on the strengths of existing approaches in political science and linguistics to accurately record information from events in a way that is standardized across different kinds of political behavior. These slots provide more information than the standard source-action-target triplet of existing political science approaches, but are much more constrained than computational linguistics models

[1] Many systems conceptually include a location slot, but techniques for properly filling location slots are only just emerging. See Halterman (2019).

that are intended to handle all possible actions.

Specifically, I propose an ontology of slots that consists of eight possible slots comprising an event.

1. An "actor" slot that the contains the actor doing the action. Grammatically, this slot will usually consist of subject nouns. In natural language processing, this slot is usually referred to as the "sender" or "agent".
2. An "action" slot that contains a description of the action that took place. Grammatically, this slot will contain at least one verb, but they also contain adverbs, adjectives, and other modifiers of the verb.
3. A "recipient" slot that contains information about the actor receiving the action. Grammatically, this lot will involve direct objects object of prepositions or indirect objects. In natural language processing this is referred to as the "receiver" or "patient" and in some earlier political science approaches (e.g. Gerner et al. (2002)), the "target".
4. An "instrument" or "means" slot, comprising the objects used by the actor in performing the action. For instance, the italicized objects in the following sentences are instruments or means: deliver *aid*, fire *mortars*, disperse using *tear gas*. Grammatically, these are reported in direct objects, prepositional phrases, and indirect objects. These grammatical roles are the same as where the action's recipient is alo reported,
5. A "reason/cause" slot for the contextual information that is often reported alongside events in political text. For instance, the italicized span in "arrested two people *for participating in last week's protests*" does not provide information about the event itself, but rather context for the event.
6. A date slot, with information on when the events took place.
7. A location slot, with information on where the events took place.
8. A "reporter" slot, with information on what source reported that the occurrence of the event.

At a minimum, an event must have an action and an actor or recipient, but other slots are optional and sentences reporting all eight pieces of information will be uncommon.

Slot filling algorithm

The most difficult step step in the process of slot filling is determining when direct objects are recipients of the action and when they are instruments of the action. (This problem sidestepped if the model only needs to locate the predicate, as opposed to distinguishing between actions and recipients (Van Atteveldt et al. 2017)). Any approach to filling the slots I specify must use both syntax (the grammar of the sentence) and semantics (the meanings of words). Figure 1 illustrates why.

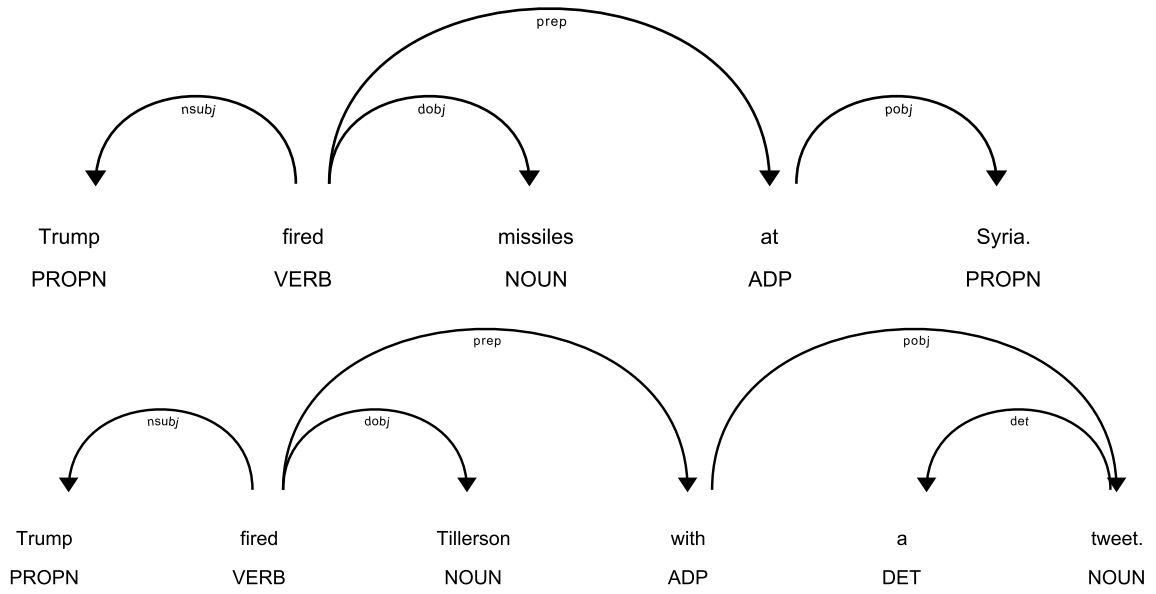

Figure 1: A dependency parse representation of two sentences. Dependency parses can be read as directed trees, beginning with a "root" verb (here, "fired") and having labeled paths (e.g. "nsubj") from parent to child nodes (e.g. "Trump", "missiles"). The all-capitals labels below each word are the words' part-of-speech tags. (Note the error made by the automated parsing system in labeling "Tillerson" as a noun instead of a proper noun.) Part-of-speech tags are invariant to the grammar of the sentence: "missiles" is a noun, but across sentences it could play the role of subject noun, direct object, dative object, or object of a preposition. The sentences are nearly identical in their grammatical structure, but the grammatical parts of the sentences correspond to different slots, illustrating the need for semantic information about the words as well. In the first example, the direct object in the sentence plays the role of an "instrument" of the action, while in the second sentence, the direct object plays the role of the recipient of the action.

A purely syntactic representation of a sentence cannot distinguish between, for instance, a direct object being an instrument of an action ("missiles") and a direct object being an actor receiving the action ("Tillerson"). In contrast, semantic analysis of words provides information about whether words are likely to describe people, actions, weapons, locations, and so on, but cannot link these words together into the meaningful relations encoded in text. My model uses both syntactic and semantic information to fill an event's slots.

My model proceeds in three steps. First, it performs a grammatical dependency parse of sentence. Next, it uses hand-specified rules to segment the sentence into rough spans. Finally it uses a machine learning model to determine whether objects are instruments/means or recipients of the action.

First, it uses the automatically-recognized dependency structure of a sentence (Honnibal and Montani 2017). Dependency parses encode the grammatical relationships between words in a sentence in a directed tree. For example, a verb ("fired") could be connected to its subject noun ("Trump") and its direct object ("Tillerson"). I generate a set of deterministic rules on this tree to produce candidate spans for the actor, action, recipient, instrument, location, date, and reporter slots for each event. *Note: the system I have implemented does not currently identify the "reason/cause" span. I added this slot to the ontology after completing an initial version of the slot filler and saw that a large number of the errors were coming from event context being coded as separate events. Future iterations of the slot filler will separate out this information into a separate slot.*

Algorithm 1

def *children*:

a word's children are the nodes immediately "below" it on the dependency tree. (E.g., in Figure 1, "tweet" is a child of "with".)

def *ancestor*:

All words upstream of the word in the dependency parse in the path to the root verb.

def *predicate subtree*:

traverse all branches of tree, with the exception of words that are marks (`mark`) or adverbial clauses (`advcl`), or words that are labeled as reporters, or words whose ancestor subject noun (`nsubj`) is different from the subject noun of interest.

input: a subject noun (`nsubj` relation)

oututs: ⟨source actor, action, recipient ⟩

1. define the actor as all the subtree[2] of the subject noun.
2. define candidate recipient spans as the subtrees of all direct objects, objects of prepositions, dative objects, and in the passive case
3. actual recipients are candidate recipient spans where the recipient labeler function returned a high predicted probability of them being recipients, rather than instruments/means.
4. the predicate is the action itself combined with the instrument/means spans (grammatically, a pruned subtree of the subject noun's parent verb, with detected recipients removed).

Note: This model is for sentences in the active voice. A slightly modified version handles passive sentences.

The syntactic information provided by the dependency parse cannot on its own fully resolve each span, however. The (grammatically identical) sentence "Trump fired missiles" uses "missiles" in a different semantic role from "Tillerson", despite their identical grammar. "Tillerson" is a recipient of the action, while "missiles" is an "instrument" of the action and belongs in the action slot alongside "fired". To resolve these issues, I train a convolutional neural network (CNN) classifier on a new set of labeled data to classify noun phrases either as recipients or instruments of actions. The model operates on the words' pretrained embeddings, meaning that it can easily classify new words it did not see during training, and the CNN can account for word order over the short spans without the computational cost of a recurrent neural network. Specifically, I create a dataset of "candidate" actors, consisting of spans of text that syntactically may be actors, but semantically may be instruments of actions. I manually labeled 2,000 of these spans. The convolutional neural network that I fit uses pretrained embeddings as inputs. Each convolution in the network is applied to a window of three words at once, meaning that the model can learn trigram information. The model stacks several convolutional layers to learn wider relationships between words. The model achieves around 85% accuracy. In production, phrases that are recognized as receivers are then removed from the predicate.

I also train a "reporter" model that recognizes phrases with a sentence that provide a source attribution for the event, such as "…, Amnesty International reported." These phrases are then removed from the sentence, preventing them from being coded as extra events, and allowing them to be added as metadata to the extracted events. The reporter recognition task is similar to named entity recognition tasks, so I use a multilayer convolutional

[2]The subtree is recursively all children of that word and the children of its children, etc.

neural network that uses pretrained word embeddings and that performs well on named entity recognition tasks (Honnibal and Montani 2017).

Van Atteveldt et al. (2017) develop a model for recognizing sources that uses a set of hand-specified rules.[3] The advantage of using a machine learning model over a rule-based system is that I achieve higher recall on actual production text.

Information on dates and locations is easily extracted using off-the-shelf named entity recognition. A more sophisticated approach to linking actions and the most specific locations where they are reported to occur is described in Halterman (2019) and could be easily incorporated into the algorithm.

Applying this slot filling technique to real world text can produce dozens of events per page, meaning that a technique for clustering or categorizing the events is needed. While supervised learning is a much more effective technique for extracting known categories from text, unsupervised learning is advantageous because it does not require labeled data and allows researchers to explore their data and learn event categories they may not have specified a priori. I therefore turn in the next section to a new unsupervised learning algorithm for clustering predicate phrases.

Aggregating actions

The second component of the model consists of learning event types by clustering extracted predicate phrases (actions plus instruments) that produce intuitive groupings of phrases. Two approaches could accomplish this. One approach is to use supervised learning to categorize text into predefined categories. This approach has the advantage of requiring analysts to pre-specify which clusters of actions are interested in, which ensures that categories are well conceptualized. Supervised models are also easily assessed accuracy in recovering their categories. They have, however, the disadvantages of requiring labeled training data and of requiring analysts to know a priori which events from the corpus are of interest.

A second approach is to used unsupervised learning to categorize events. Unsupervised learning has the advantages of allowing analysts inductively learn events from text and does not require label training data. It comes at the cost, however, of producing categories that may not have the meaning that analysts ascribe to them.[4] Unsu-

[3] I use the term "reporter" instead of "source" to avoid confusion with the terminology used in other event extraction literature, where "source" is often used where I use the term "actor".

[4] A hybrid approach combines unsupervised clustering with the small number of human analyst decisions. For example, Ritter et al. (2015) propose a weakly supervised model for recognizing events, that require analysts to only specify small number of positive documents of interest. This approach is potentially promising for this application but is left for future work.

pervised techniques, though, have become the dominant approaching to analyzing text in political science.

Although the slot filling model return spans for both actors and events, I focus on clustering events as opposed to actors. Researchers often have better a priori ideas about the identity of actors as opposed to events, as people, organizations, and governments have fairly well delineated boundaries between them, while categories of events are more continuous and more subjectively defined. This creates a greater need to inductively learn events from text than actors. Second, actors are easier to categorize using dictionary or supervised machine learning methods. It is easier to specify a list of words training machine learning classifier to recognize specific people or organizations or governments then it is to enumerate every way of describing the particular kind of behavior.

I introduce a new method of topic modeling that is optimized for very short (even single-word) documents and that is iteratively fit using an EM algorithm on pretrained word embeddings. I describe the desirable behaviors in a text clustering algorithm, explain why existing approaches do not work on short documents, and propose my own model.

A good unsupervised event aggregation system should have several properties. First, because it is inductively learning events from text, it clearly cannot depend on a pre-specified grouping of event types. This requirement distinguishes it from previous approaches to event extraction in clinical science and much of the semantic role labeling literature in natural language processing. Second, in order to be useful to applied researchers, the model needs to generalize well across different domains. It should work on multiple text types, including newspapers, newswires, and government documents, as well as on a wide variety of event types ranging from political mobilization and political violence two economic events. This will ensure that the cost of producing new data remains low. Third, the model needs to incorporate outside, prior information on the meanings of words. This is because short documents and small corpora make it difficult to learn the meanings of words from scratch with every set of documents. As much as possible, the model should approach documents in the same way as a human reader, applying prior information about the meanings of words learnt from exposure to a large body of text. Finally, in the context of clustering actions the model should incorporate grammatical information to determine which words are more important to discerning the meaning of the action in other words.

The standard approach to learning clusters of documents in political science is using latent Dirichlet allocation (Blei, Ng, and Jordan 2003) or one of its variants (Blei and Lafferty 2007; Roberts et al. 2013). For very short "documents" of the kind produced in slot filling tasks, several assumptions of these models break down in significant ways. I offer a short intuitive explanation of why this is the case. First, LDA assumes that each word's probability in a document depends solely on its topic indicator: $p(w_i|z_i) \sim \text{Mult}(\beta_z)$. This assumption is clearly a simpli-

fication, as the occurrence of words in a document changes the probability of seeing other words, beyond the information contained in the document's topic proportions. In short documents, especially very grammatically constrained sentences, the negative correlation induced between words becomes severe: in most grammatical constructions of a phrase, only a single verb will be present, meaning that the presence of a single verb induces complete negative correlation between all verbs. For less constrained parts-of-speech, such as adjectives, the negative correlation is no longer absolute, but synonyms are unlikely to co-occur in very short text. In extreme cases, the extracted phrases are single words. Some limited work exists on using LDA for very short documents. For example, Ritter et al. (2012) use LDA for learning events from Twitter, but focus on proper names and dates rather than the set of all words.

The solution to the short document problem emulates how a human analyst would solve the problem, by drawing on on previous knowledge of the meanings of words in wider context to group events together. The challenge then becomes representing sentences in a way that incorporates prior information.

Sentence Embedding

All automated text analysis requires making decisions about how to represent text in numeric form in order to process it. For mathematical convenience, this generally requires representing variable-length text as a fixed length input.[5] I represent each span as a function of the span's individual word embeddings to produce a "span embedding". Word embeddings are a standard technique in natural language processing for representing words. Embeddings generally represent words as short, dense vectors, where words that are used in similar contexts will have vector representations with high similarity. In addition to representing words more efficiently, embeddings can also be used to incorporate prior knowledge about the meaning and usage of words, reducing the training data needed in specific applications and sharing information across tasks.[6] Pretrained embeddings are available, produced on very large corpora of cross domain text. A common approach to producing sentence- or document embeddings is to simply average (elementwise) the sentence's constituent embeddings. Averaging has the disadvantages of treating all words as equally informative, and has the effect of pushing the vector toward the corpus mean as the length of the document increases. As an atheoretical correction, some models concatenate the elementwise maximum of the words' vectors to the averaged vector as a secondary input.

[5]The major exception is recurrent neural networks, which can accept variable length text, though they still require words or letters to be represented as fixed length inputs.

[6]Word embeddings are thus a basic form of transfer learning, where models trained on one task can be used as a starting point for models being trained in a new domain. After years of success using transfer learning in image recognition, natural language processing has had a explosion of transfer learning models in the past two years (Howard and Ruder 2018; Peters et al. 2018; Devlin et al. 2018). See Ruder (2018) for a non-technical overview.

Instead I adopt a sentencing embedding model proposed by Arora, Liang, and Ma (2017). Their model is theoretically motivated and simple to implement and also achieves very good performance on sentence classification tasks, beating even sophisticated supervised sentence classification models. Their sentencing embedding is a weighted transformed mean of the pretraining word embeddings. Let v_w be the pretrained word embedding and $p(w)$ be the empirical frequency of word w in a large corpus. Each sentence embedding is initially represented using a smoothed, weighted elementwise mean of its constituent words' embeddings:

$$\tilde{v}_i = \frac{1}{|d|} \sum_{w \in d} \frac{a}{a + p(w)} v_w, \tag{1}$$

where $a = 0.0001$ is a smoothing hyperparameter. This weighting approximates the standard tf-idf weighting scheme in traditional text analysis and information retrieval. The sentence vectors then have a "common component" removed, whereby the first singular vector of all the vectors in the corpus are removed from each: $v_i = \tilde{v}_i - uu^T \tilde{v}_i$, where u is the first singular vector of the matrix X of all $\tilde{v}_i$. Arora, Liang, and Ma (2017) propose a generative model of text, whereby words are emitted at each time step based on their embedding's similarity to a latent "discourse vector" in the same space as the word embeddings. The sentence embedding they produce approximates the maximum likelihood estimate of this discourse vector.

Word embeddings and latent Dirichlet allocation have a mathematical connection. LDA is a probabilistic factoring of a matrix of counts of words in each document into a document-topics matrix ($\boldsymbol{\theta}$) and a topic-words matrix ($\boldsymbol{\beta}$) (Buntine 2002; Hoffman, Bach, and Blei 2010). word2vec (Mikolov et al. 2013), one of the common word embedding algorithms, is a factoring of the pointwise mutual information matrix of words and contexts (Levy and Goldberg 2014). (In this application, with spans as documents, the word2vec window would encompass the entire "document"). A major difference is that LDA is run on just the corpus at hand, while the embeddings are pre-trained. Human readers don't arrive at a document unsure if "arrest" is more similar to "detain" or "citizen", but LDA does.

I modify the SIF sentence embedding model to vary word weights by their part-of-speech, in addition to by their word frequency. Because I focus on the specific domain of actions in the clustering algorithm, and because verbs are generally the most important component, I give their embeddings full weight.[7]

[7]Other weighting schemes are possible. Specifically, the direct objects ("instruments") of actions could play a major role in determining where in latent event space an action belongs, and therefore receive high weights as well. Future work could empirically establish good values for weighting functions by estimating the effects of different weights on downstream classification performance. Specifically, given 1,000–2,000 spans manually labeled with categories of event types, a classifier could be trained on SIF embeddings using different part-of-speech weights. The higher performing weights could be used in the "production" version.

Clustering

Clustering is a process for learning a useful, low-dimensional representation of data by placing "similar" units closer than "different" units in some space. The decisions involved in designing clustering involve picking definitions of distance and space that produce clustering results that are useful to researchers in some way, similar to the decisions involved in designing clustering algorithms for international relations (Zhukov and Stewart 2013).

When inductively learning clusters, the definitions of clusters and the membership of units in the clusters depend on each other. In other words definition of a cluster depends on its members but membership in a cluster requires a prior definition of clusters. I resolve this self-dependence problem by iteratively updating cluster membership and cluster definition using the standard approach of expectation-maximization (EM). The intuition of my EM model is similar to Gaussian mixture models or any similar EM clustering algorithm: iteratively improve the model that assigns observations to clusters, weighting units by how well each model predicts their label.

I introduce the following algorithm for inductively learning events from text:

Algorithm 2

Inputs: documents $i = 1...D$

Outputs: w_{ik}: cluster membership probability for cluster $k = 1...K$ for all documents $i = 1...D$.

θ_k, parameters describing the cluster membership model for cluster k.

Procedure:

1. Calculate the span representation v_i for all i using the SIF embedding method
2. initialize cluster label y for K topics using k-means in the space of embedded phrases and continue to the M step.
3. until convergence do:

- E Step

  For each span $i \in D$:

  - Calculate the cluster membership probability:

    $w_{ik} = \text{logit}^{-1}\left(v_i^T \theta_k\right)$

- Assign the maximum predicted label to each document:

  $y_i = \text{argmax}(\{w_{i1}, ...w_{ik}\})$

- M Step

  For each cluster $k \in K$, fit a weighted logistic regression:

  - $\theta_k = \text{argmax}_{\theta_k} \prod_{i \in D} p_{ik}^{\mathbb{1}(y_i = k)} (1 - p_{ik})^{\mathbb{1}(y_i \neq k)} w_{ik}$,
    where $p_{ik} = \text{logit}^{-1} \left(v_i^T \theta_k\right)$

At a high level, the clustering algorithm assumes that each cluster is defined as a region in embedding space defined by by a transformed linear function of embedding space. Weights are provided by the models' accuracy in determining whether a point belongs inside or outside cluster k.

Topic modeling evaluation

To evaluate the topic models, I first assess their accuracy on a set of synthetic "documents" created by a known data generating process. While evaluation of methods against simulated data is a standard technique in most quantitative methodology, it is rarely applied in in topic modeling (though see Boyd-Graber and Blei 2009). I manually specify verbs, direct objects, and adjectives corresponding to eight political topics (see Appendix B). Each document is generated by sampling a topic indicator, then a single corresponding verb from that topic's set of verbs, and 0-4 other words for that topic. For example, one topic in the simulation contains "meeting" words, including the verbs "deliver" and "provide" and other predicate words "humanitarian", "water", "food", and "aid". Generated documents also include draws from a set of "junk" terms consisting of conjunctions and prepositions that are shared across all topics.

I compare the performance of my model with a standard LDA model across several document sizes, performing 100 simulations for each condition.

Although the differences in performance are relatively small on the synthetic data set, the embedding prior model seems to perform somewhat better than LDA. Future work is needed to assess the differences in performance on the real data sets. Anecdotally, however, the embedding prior model yields topics that are more semantically coherent than those produced by LDA on very short spans/documents.[8]

[8] See appendix A for an aside discussion.

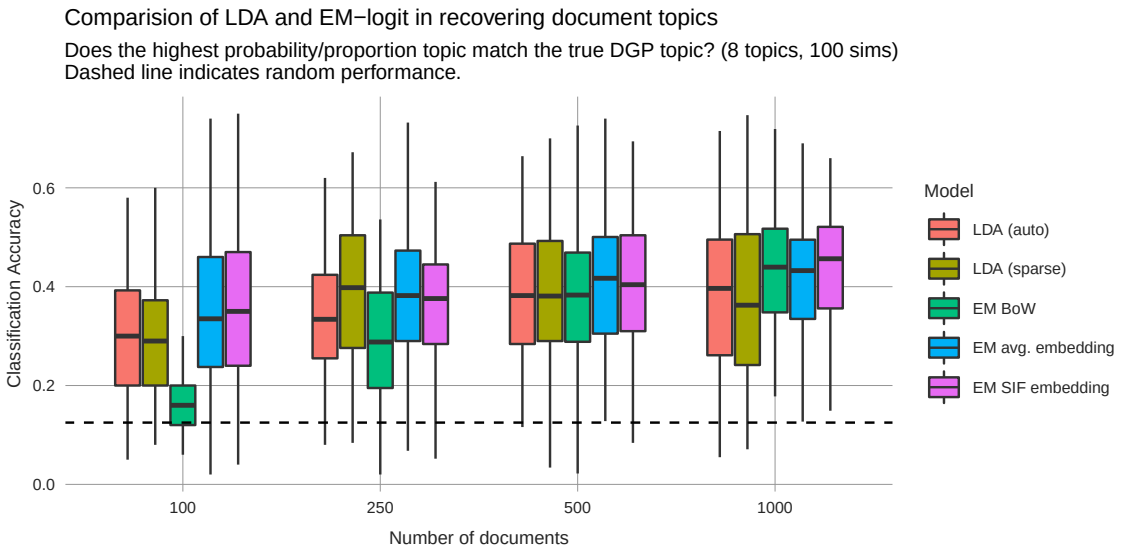

Figure 2: comparing LDA and the iterative embedding prior topic model on synthetic data. The iterative embedding prior topic model performs someone better than LDA, although differences are small on the synthetic data set. The auto LDA model uses a self-tuning hyperparameter for the expected number of topics per document. The sparse model is set to find one topic per document which matches the data generating process.

Application: Has respect for human rights increased over time?

An ongoing debate in international relations and comparative politics concerns whether respect for human rights has changed over time. Many observers expect, on anecdotal or qualitative grounds, that the global human rights situation has improved since the 1970s. In contrast, the major datasets of respect for human rights, including the CIRI Human Rights Dataset (Cingranelli and Richards 2004) and the Political Terror Scale (Wood and Gibney 2010) dataset show a fairly constant level of human rights violations over the past four decades.

Fariss (2014; 2018) argues that this counterinuitive finding is the product of changes in how human rights violations are reported. As NGOs gain greater access and human rights observers have better information, a greater proportion of human rights violations will be recorded than in the past. If the probability of detecting human rights violations is increasing faster than the overall rate of actual violations is decreasing, we will observe an apparent increase in human rights violations. Similarly, as the human rights record improves in different countries, human rights activists are likely to change the focus of their activism to other, less egregious violations.

Fariss (2014) models this change using a dynamic IRT model, using incidents of genocide as a perfectly observed anchoring observation to estimate the probability of incident reporting. He distinctions between what he calls "event" and "standards"-based reporting, with "events" like genocide being more accurately measured

than the "standards" that the State Department and Amnesty International measure because the definition of events changes less than the definition of standards and because data on events is updated retrospectively as better information becomes available.

The paper makes an important contribution to the debate in positing the existence and mechanisms of changing reporting standards. The model that it uses, however, rests on several major assumptions, the greatest of which is that all state repression, from arbitrary arrest to genocide, exists along a single latent space, meaning that values can be compared across them. Instead, we might believe that genocide is simply different from other violations of human rights, violating the assumption of the unidimensional latent variable. D. Cingranelli and Filippov (2018a) and D. Cingranelli and Filippov (2018b) dispute this finding, largely on objections to Fariss's IRT model.

Rather than relying on the same limited set of country-year ratings to measure human rights respect and the changing standard of human rights violations, I instead generate new data on respect for human rights by returning to the original State department text used to create the country year ratings. Other researchers (Greene, Park, and Colaresi 2019) have also begun looking directly at the text, but in ways that do not preserve the relationships between actors and actions in the text.

I applied both steps of my new method to the State Department's annual country human rights reports from 1977 until 1999, when the format of the documents changed. From this text, the event extraction model produced 1.02 million events. Because this debate is over government respect for human rights, I then subset the events to only those in which the extracted actor span contained terms in a list of terms that I specified. This list included all country names and demonyms, along with terms describing government officials, such as "soldier", "authorities", "police", or "government". Approximately one quarter of the total events, 243,449, had actor spans that included these words. The date I produce is thus a compromise between between what Fariss calls standards-based reporting and event reporting. Rather than producing a single country or score as in the standard approach I produce a set of disaggregated events. Unlike codings of genocide, however, these machine extracted events are not updated retroactively as better data becomes available.

I then fit the embedding prior iterative topic model to these extracted stands using the SIF embedding method using 60 topics. 60 was the greatest number of topics I tried and produced the best results. Many of the learns topics are specific and contain only a small number of spans. A small number of topics together contain the majority of spans, which may be better modeled by an even larger number of topics.

Empirical Results

As Fariss observes, the total amount of reporting, measured by the number of words, has increased over time. Figure 3 shows that the number of reported events is gone up as well, from approximately 2,500 per year to around 25,000 per year. On its own, this figure offers some suggestive evidence that the standard of reporting has changed. We may believe that human rights practices are stagnant or perhaps even slightly worsening, but we do not believe that human rights violations have become an order of magnitude more common. Moreover, the event density of the reporting has changed as well. From 1979 to 1999, the number of events has gone from 10 to 16 events per 1,000 words. This indicates at least higher specificity in the content of the reports. Interestingly however the proportion of events with government actors remains steady between 22% to 25% over the period.

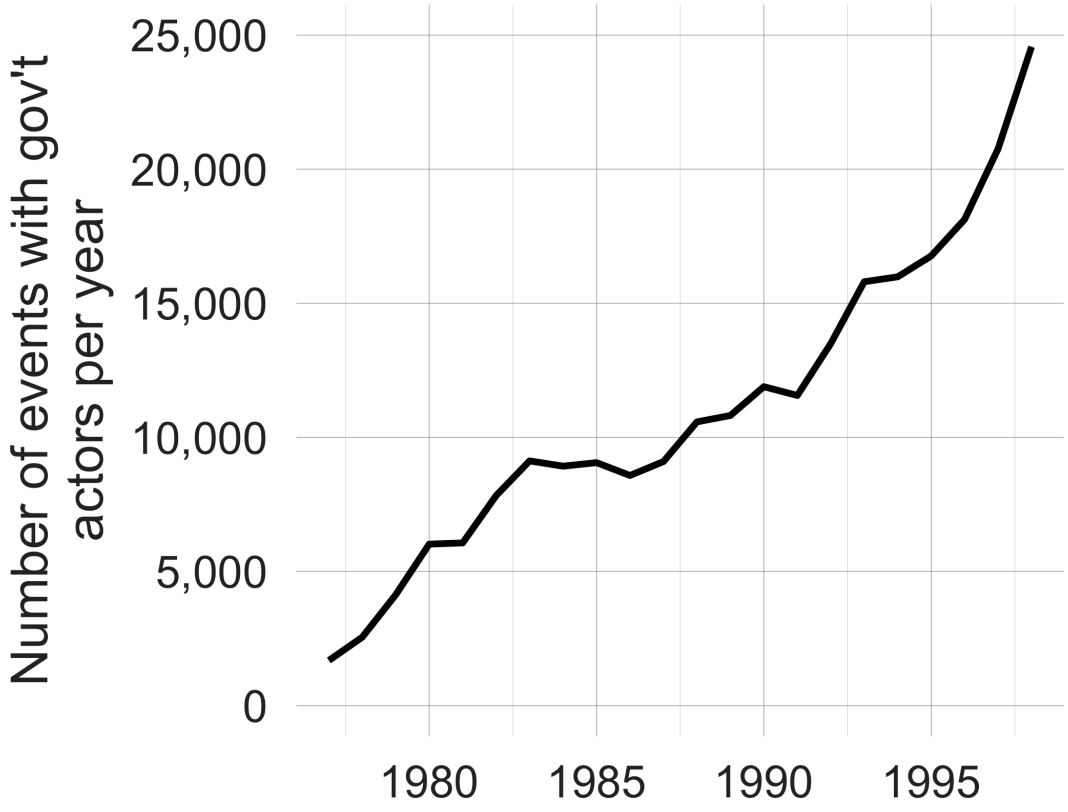

Figure 3: Change in proportion of total events by event type.

The real strength of the new method comes in learning different events types rather than looking only with the aggregate total. When we decompose the total line into a proportion of each event type we see variation in which

event type are occupying a larger proportion of total events (Figure 4).

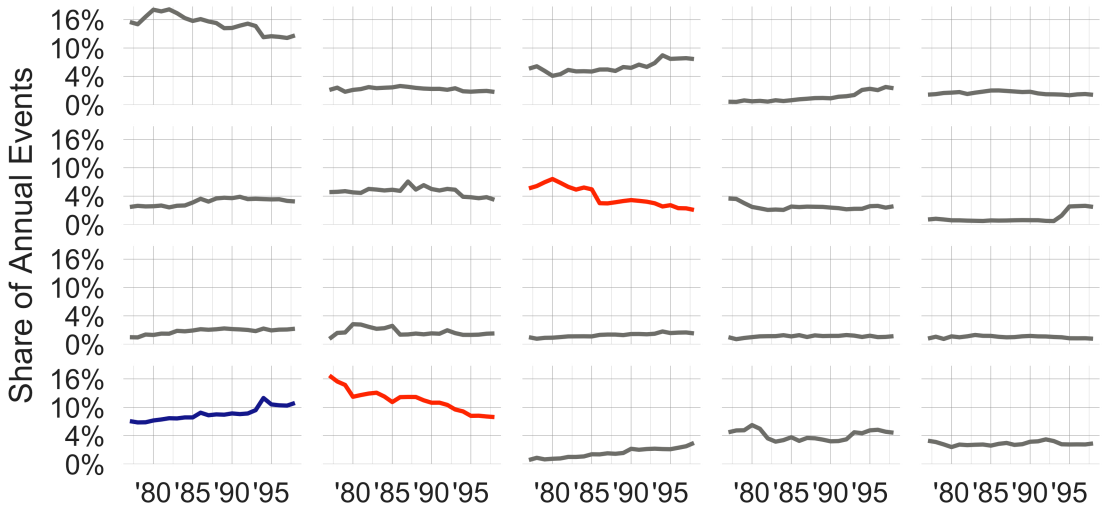

Figure 4: Change in proportion of total events by event type.

If we then focus only on the topics whose shares are decreasing, we again see suggestive evidence for changing standard of reporting. Inspecting high probability documents for these clusters indicates that three of these clusters described not human rights abuses but other events or information about these countries (Figure 5). Topic 16 seems to focus on descriptions of the countries economic system. Topic 47 includes in large part positive reports of the countries respect for rights. Topic 57 is a fairly uncommon topic that reports information about countries regime type. The fact that these three topics are declining and there proportion offer some evidence that the standard of reporting has changed.

Conclusion

This paper introduces two new techniques that together allow researchers to inductively learn political events from text. A slot filling model uses grammatical information and new machine learning models to identify the parts of a sentence corresponding to different "slots" in an event. It does so with much finer resolution than previous grammar-based event extraction models, and with far greater coverage than dictionary based methods. A second model takes these short spans and aggregates them into useful categories for further analysis. It overcomes the short document problem by using prior information in the form of word embeddings and an iter-

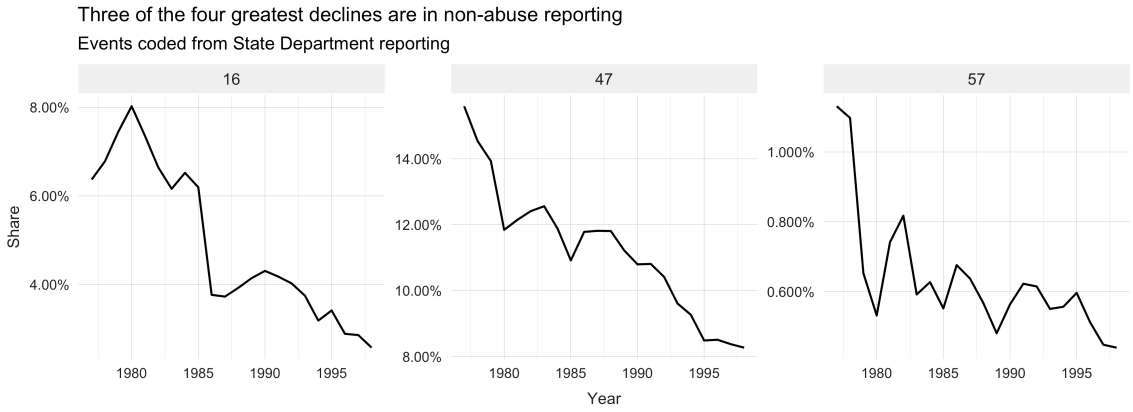

Figure 5: Change in proportion of total events by event type.
16: "made internal economic reform and market stabilization but found", "pursued a successful export-oriented agricultural growth strategy", "own and run banking and insurance, air and rail transport, public utilities, and key heavy industries".
47: "After considerable criticism and protest withdrew the bill",
"achieved its independence under", "exerts influence in some cases", "are free express their opinions privately and, to a certain extent", "provides for certain basic rights".
57: "is a constitutional monarchy", "have lively and free, -, multiparty political systems", "is a multiparty democracy with mandatory universal suffrage", "is a representative democracy".

ative EM model to simultaneously learn cluster definitions and cluster membership. This model potentially has broader applicability beyond event extraction. It could be useful in other situations where very short documents need to be clustered. I then apply the model to an open question in international politics, about whether respect for human rights as improved overtime. I produce new disaggregated data on human rights related events with government actors and offer some evidence for the arguments that the standard of reporting has changed over time. While the volume of human rights reporting has increased greatly over time, specific kinds of rights violations have changed in their overall proportion of reporting. Because the model is completely general it can be applied to a wide range of questions in political science, anywhere information on the behavior of actors is important.

References

Althaus, Scott L, Buddy Peyton, and Dan A Shalmon. 2018. "Spatial and Temporal Dynamics of Boko Haram Activity in 6 Event Data Pipelines." *APSA Mini Conference on Modern Event Data Development and Analysis*.

Arora, Sanjeev, Yingyu Liang, and Tengyu Ma. 2017. "A Simple but Tough-to-Beat Baseline for Sentence Embeddings." *ICLR*.

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.

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

Blei, David M, and John D Lafferty. 2007. "A Correlated Topic Model of Science." *The Annals of Applied Statistics* 1 (1): 17–35.

Blei, David M, Andrew Y Ng, and Michael I Jordan. 2003. "Latent Dirichlet Allocation." *Journal of Machine Learning Research* 3 (Jan): 993–1022.

Boschee, Elizabeth, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. "ICEWS Coded Event Data." *Harvard Dataverse* 12.

Boyd-Graber, Jordan L, and David M Blei. 2009. "Syntactic Topic Models." In *Advances in Neural Information Processing Systems*, 185–92.

Buntine, Wray. 2002. "Variational Extensions to Em and Multinomial Pca." In *European Conference on Machine Learning*, 23–34. Springer.

Chang, Jonathan, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber, and David M Blei. 2009. "Reading Tea Leaves: How Humans Interpret Topic Models." In *Advances in Neural Information Processing Systems*, 288–96.

Cingranelli, David L, and David L Richards. 2004. "CIRI Human Rights Dataset." http://www.humanrightsdata.com.

Cingranelli, David, and Mikhail Filippov. 2018a. "Are Human Rights Practices Improving?" *American Political Science Review* 112 (4). Cambridge University Press: 1083–9.

———. 2018b. "Problems of Model Specification and Improper Data Extrapolation." *British Journal of Political*

Science 48 (1). Cambridge University Press: 273–74.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." *arXiv Preprint arXiv:1810.04805*.

Dowty, David. 1991. "Thematic Proto-Roles and Argument Selection." *Language* 67 (3). Linguistic Society of America: 547–619.

Fariss, Christopher J. 2014. "Respect for Human Rights Has Improved over Time: Modeling the Changing Standard of Accountability." *American Political Science Review* 108 (2). Cambridge University Press: 297–318.

———. 2018. "Are Things Really Getting Better? How to Validate Latent Variable Models of Human Rights." *British Journal of Political Science* 48 (1). Cambridge University Press: 275–82.

Geddes, Barbara, Joseph Wright, and Erica Frantz. 2014. "Autocratic Breakdown and Regime Transitions: A New Data Set." *Perspectives on Politics* 12 (02): 313–31.

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*.

Goemans, Henk E, Kristian Skrede Gleditsch, and Giacomo Chiozza. 2009. "Introducing Archigos: A Dataset of Political Leaders." *Journal of Peace Research* 46 (2). Sage Publications Sage UK: London, England: 269–83.

Greene, Kevin T, Baekkwan Park, and Michael Colaresi. 2019. "Machine Learning Human Rights and Wrongs: How the Successes and Failures of Supervised Learning Algorithms Can Inform the Debate About Information Effects." *Political Analysis* 27 (2). Cambridge University Press: 223–30.

Halterman, Andrew. 2019. "Geolocating Political Events in Text." *NLP+CSS Workshop, NAACL*.

Hoffman, Matthew, Francis R Bach, and David M Blei. 2010. "Online Learning for Latent Dirichlet Allocation." In *Advances in Neural Information Processing Systems*, 856–64.

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

Howard, Jeremy, and Sebastian Ruder. 2018. "Universal Language Model Fine-Tuning for Text Classification." *arXiv Preprint arXiv:1801.06146v2*.

Jones, Daniel M, Stuart A Bremer, and J David Singer. 1996. "Militarized Interstate Disputes, 1816–1992: Ra-

tionale, Coding Rules, and Empirical Patterns." *Conflict Management and Peace Science* 15 (2): 163–213.

LaFree, Gary, and Laura Dugan. 2007. "Introducing the Global Terrorism Database." *Terrorism and Political Violence* 19 (2): 181–204.

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

Makarov, Peter. 2018. "Automated Acquisition of Patterns for Coding Political Event Data: Two Case Studies." In *Proceedings of the Second Joint Sighum Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, 103–12.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. "Distributed Representations of Words and Phrases and Their Compositionality." In *Advances in Neural Information Processing Systems*, 3111–9.

Nardulli, Peter F, Scott L Althaus, and Matthew Hayes. 2015. "A Progressive Supervised-Learning Approach to Generating Rich Civil Strife Data." *Sociological Methodology* 45 (1): 148–83.

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

O'Connor, Brendan, Brandon Stewart, and Noah A Smith. 2013. "Learning to Extract International Relations from Political Context." *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* Vol. 1.

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.

Pavlick, Ellie, Travis Wolfe, Pushpendre Rastogi, Chris Callison-Burch, Mark Dredze, and Benjamin Van Durme. 2015. "FrameNet+: Fast Paraphrastic Tripling of Framenet." In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2:408–13.

Peters, Matthew E, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. "Deep Contextualized Word Representations." *arXiv Preprint arXiv:1802.05365*.

Powell, Jonathan, and Clayton Thyne. 2011. "Global Instances of Coups from 1950 to 2010: A New Dataset."

Journal of Peace Research 48 (2): 249–59.

Raleigh, Clionadh, Andrew Linke, Håvard Hegre, and Joakim Karlsen. 2010. "Introducing ACLED: An Armed Conflict Location and Event Dataset: Special Data Feature." *Journal of Peace Research* 47 (5): 651–60.

Ritter, Alan, Oren Etzioni, Sam Clark, and others. 2012. "Open Domain Event Extraction from Twitter." In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1104–12. ACM.

Ritter, Alan, Evan Wright, William Casey, and Tom Mitchell. 2015. "Weakly Supervised Extraction of Computer Security Events from Twitter." In *Proceedings of the 24th International Conference on World Wide Web*, 896–905. International World Wide Web Conferences Steering Committee.

Roberts, Margaret E, Brandon M Stewart, Dustin Tingley, Edoardo M Airoldi, and others. 2013. "The Structural Topic Model and Applied Social Science." In *Advances in Neural Information Processing Systems Workshop on Topic Models: Computation, Application, and Evaluation*.

Ruder, Sebastian. 2018. "NLP's Imagenet Moment Has Arrived." *The Gradient* https://thegradient.pub/nlp-imagenet/ (July).

Rudinger, Rachel, and Benjamin Van Durme. 2014. "Is the Stanford Dependency Representation Semantic?" In *Proceedings of the Second Workshop on Events: Definition, Detection, Coreference, and Representation*, 54–58.

Salehyan, Idean, Cullen S Hendrix, Jesse Hamner, Christina Case, Christopher Linebarger, Emily Stull, and Jennifer Williams. 2012. "Social Conflict in Africa: A New Database." *International Interactions* 38 (4): 503–11.

Schrodt, Philip A. 2009. "TABARI: Textual Analysis by Augmented Replacement Instructions." *Dept. of Political Science, University of Kansas, Blake Hall, Version 0.7. 3b3*, 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): 561–87.

Sundberg, Ralph, Kristine Eck, and Joakim Kreutz. 2012. "Introducing the UCDP Non-State Conflict Dataset." *Journal of Peace Research* 49 (2): 351–62.

Van Atteveldt, Wouter, Tamir Sheafer, Shaul R Shenhav, and Yair Fogel-Dror. 2017. "Clause Analysis: Using Syntactic Information to Automatically Extract Source, Subject, and Predicate from Texts with an Application

to the 2008–2009 Gaza War." *Political Analysis* 25 (2). Cambridge University Press: 207–22.

White, Aaron Steven, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. "Universal Decompositional Semantics on Universal Dependencies." In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1713–23. Austin, Texas: Association for Computational Linguistics.

Wood, Reed M, and Mark Gibney. 2010. "The Political Terror Scale (Pts): A Re-Introduction and a Comparison to Ciri." *Human Rights Quarterly* 32 (2): 367–400.

Zhukov, Yuri M, and Brandon M Stewart. 2013. "Choosing Your Neighbors: Networks of Diffusion in International Relations." *International Studies Quarterly* 57 (2). Wiley Online Library: 271–87.

Appendix A: Better Topic Model Evaluations

Aside for Rich: Here we are back at the old question how to evaluate topic quality. I like this approach of generating simulated data, but I think the topics need more words in order to be realistic. I've also abandoned my earlier efforts to measure whether words are being assigned to the right topics, in part because my method does not operate at the level of words.

I would also like to conduct human evaluations on the quality of topics learned from real text. One way I've been thinking about doing this is to have annotators evaluate topic quality for me. I would show annotators two topics side-by-side in random order, one generated using my method and one generated using LDA. Each topic would consist of 5ish randomly selected "documents" (short spans) from that topic. Annotators would then be asked which topic displays greater coherence. By collecting enough of these annotations I could determine whether LDA or my method generally produced better topics.

The other approach that I think is promising is an "intrusion" task (Chang et al. 2009). With very short documents like the spans that I have, it's feasible to do a document intrusion task, rather than the word intrusion tasks that Chang et al. (2009) do. I saw a "tweet intrusion" task at NAACL this year that seemed to work pretty well. It could even be possible to get at "usefulness" of the topics more directly by giving experimental participants a social science question and asking them which topics are more useful for them in answering the question.

(end aside)

Appendix B: Breaking the Fourth Wall

Hi Rich,

Now that we're at the end of the paper, I'd like to address a few questions to you specifically, outside the flow of the paper.

Weighting observations

The first question I've had is about how to correctly weight observations in the EM algorithm. Recall that the weights are needed to account for existing cluster membership when updating the cluster parameters in the M step. There are two approaches I've considered. I've decided on the latter after some back and forth (this is the

weighting present in the paper) and I'd like your thoughts on whether you think that was the right decision, and whether you can think of what the appropriate justification is.

The first weighting approach is a standard EM clustering approach to weighting (think Gaussian mixture models), where the units used to generate a cluster's parameters are weighted by their probability of membership in the cluster. Formally, the weight for unit i is given by

$$w_i = p(z_i = k), \tag{2}$$

where z_i is the cluster label for unit i, and $k \in K$ is the topic/cluster number. Thus, parameter updates are something like $\theta = \text{argmin}_\theta \sum_i \sum_k w_{ik} \text{Loss}(\theta, z_i)$. This is what GMMs and the other EM clustering algorithms I've seen do. It seems strange here, though. Units that are assigned to a cluster receive high weights in the regression to predict cluster membership, whereas units that are not assigned to a cluster receive low weight in the regression. This leads to the strange result where all the 1s have high weight, and all the 0s have low weight. This might explain the tendency of the largest cluster to keep growing in some of the previous models that I ran.

The second approach, which makes more sense to be but is less precedented, is to weight by the confidence the model has in cluster assignment:

$$w_{ik} = p(z_i = k)^{z_i}(1 - p(z_i = k))^{(1-z_i)}. \tag{3}$$

This would assign high weights to observations where the model is very certain it either belongs inside or outside the cluster, and low weights to where the model is uncertain. This approach makes more sense but is not the standard one in EM clustering. (An idle observation: this is basically the inverse of boosting. The model is increasingly weighting the observations it gets right, rather than the ones it gets wrong). Do you think I made the right decision in opting for the second weighting scheme? Do you think there's a reasonable justification for it?

"Dirichlet"/categorical regression

I've been considering a different approach to modeling topics that would both be cleaner than a series of logistic regressions and would better preserve a mixed membership quality to the topics. (Note: I think that it's *okay* to

treat these short spans as single topics, not that it's necessarily the most desirable way to). Here's the beginnings of a formalization of how a categorical clustering regression would work.

Consider a span of text $i = 1...D$ represented as a vector of data x_i, and a vector of cluster membership probabilities ("topic proportions") $\boldsymbol{z}_i \in \mathbb{R}^K, \sum_{k=1}^{K} z_{ik} = 1$. Thus, $\boldsymbol{z}_i$ is a vector in the $k-1$ simplex. x_i could be a word embedding or a bag of words vector but the precise details aren't important.

We assume that z_{ik}, the proportion of document i that is in cluster k is a generalized linear function of its vector representation:

$$z_{ik} = g\left(x_i^T \theta_k\right). \tag{4}$$

Following common practice in the machine learning literature, we can use the softmax function for g, resulting in

$$z_{ik} = \frac{\exp\left(x_i^T \theta_k\right)}{\sum_{k'=1}^{K} \exp\left(x_i^T \theta_{k'}\right)} \tag{5}$$

The function could be further extended by adding a tunable constant to each exponent component:

$$z_{ik} = \frac{\exp\left(\alpha \cdot x_i^T \theta_k\right)}{\sum_{k'=1}^{K} \exp\left(\alpha \cdot x_i^T \theta_{k'}\right)}, \tag{6}$$

where small values of α produce less extremized values, and large values of α push values toward extremes. This parameter is often referred to as "temperature" in the machine learning literature, but also has a Bayesian interpretation related to pseudocounts in a prior distribution.

In practice, because cluster memberships and cluster parameters depend mutually on each other, both would be estimated through in iterative process of t timesteps.

E step:

- estimate cluster membership:

$$z_{ik}^{(t)} = \frac{\exp\left(x_i^T \theta_k^{(t-1)}\right)}{\sum_{k'=1}^{K} \exp\left(x_i^T \theta_{k'}^{(t-1)}\right)} \quad \forall k \tag{7}$$

- estimate weights:

Let's provide weights related to the inverse variance of each predicted cluster membership, normalized to be between zero and one:

$$w_{ik} = 1 + \frac{-z_{ik}(1 - z_{ik})}{0.25}, \tag{8}$$

which gives us a nice curve from $w_{ik} = 1$ at $z_{ik} = 0, 1$ to $w_{ik} = 0$ at $z_{ik} = 0.5$.[9]

M step:

Given the weights and topic assignments from the E step, update the parameter values for the function.

$$\begin{aligned}
\theta^{(t)} &= \underset{\theta}{\text{argmin}}\ \text{Loss}(\theta, X_i) \\
\theta^{(t)} &= \underset{\theta}{\text{argmin}} \sum_{i=1}^{D}\sum_{k=1}^{K} w_{ik}\left(z_{ik} - z_{ik}^{(t-1)}\right)^2, \text{ subject to } \sum_{k'=1}^{K} z_{ik} = 1 \\
&\quad \text{Using our definition from above, and recognizing that it imposes the necessary constraint on } z_i, \\
&= \underset{\theta}{\text{argmin}} \sum_{i=1}^{D}\sum_{k=1}^{K} w_{ik}\left(\frac{\exp\left(x_i^T\theta_k\right)}{\sum_{k'=1}^{K}\exp\left(x_i^T\theta_{k'}\right)} - z_{ik}^{(t-1)}\right)^2
\end{aligned} \tag{9}$$

To minimize this quantity, we begin by taking the derivative:

$$\begin{aligned}
\frac{\partial \text{Loss}}{\partial\theta} &= \frac{\partial}{\partial\theta}\sum_{i=1}^{D}\sum_{k=1}^{K} w_{ik}\left(\frac{\exp\left(x_i^T\theta_k\right)}{\sum_{k'=1}^{K}\exp\left(x_i^T\theta_{k'}\right)} - z_{ik}^{(t-1)}\right)^2 \\
&= \sum_{i=1}^{D}\sum_{k=1}^{K} 2w_{ik}\left(\frac{\exp(x_i^T\theta_k)}{\sum_{k'}\exp(x_i^T\theta_{k'})} - z_{ik}^{(t-1)}\right)\cdot\frac{\exp(x_i^T\theta_k)}{\sum_{k'}\exp(x_i^T\theta_{k'})}\cdot\frac{\sum_{k'}\exp(x_i^T\theta_{k'}) - \exp(x_i^T\theta_k)}{\sum_{k'}\exp(x_i^T\theta_{k'})}\cdot \\
&\quad \frac{\exp(x_i^T\theta_k)x_i^T}{\sum_{k'}\exp(x_i^T\theta_{k'})x_i^T}
\end{aligned} \tag{10}$$

Setting to zero and analytically finding the minimum value for θ_k is infeasible because of the sums of exponents in the denominator. Instead, we can use the derivative to implement a gradient descent algorithm.

[9] Question to myself: does this need to be adjusted to the base rate? So if there are 20 categories, the base rate bet is 0.05, not 0.5?? And does $\sum_{k=1}^{K} w_{ik} = 1$?

$$\theta^{(t)} = \theta^{(t-1)} - \eta \nabla \text{Loss}\left(\theta^{(t-1)}\right), \tag{11}$$

where η is a learning rate parameter. This is standard practice in the machine learning literature when analytically minimizing the loss function is intractable.

Questions:

– do you agree that this approach conceptually makes more sense than the single membership regression models?

- does this help with demonstrating “math virtuosity”?

Appendix B

```
    verbs" :  0 : "shoot kill fire attack"
              1 : "assist help provide bring"
              2 : "meet talk discuss"
              3 : "gather protest demonstrate carried march"
              4 : "capture abduct"
              5 : "deliver provide"
              6 : "seize take capture overrun recapture"
              7 : "occupy control"


 "objects" :  0 : "village town villager civilian militant"
              1 : "aid convey help village town development government"
              2 : "karzai ambassador embassy"
              3 : "demonstration chant opposition near "
              4 : "aid worker civilian innocent"
              5 : "humanitarian water food aid"
              6 : "base territory village control area stronghold"
              7 : "base territory village area stronghold"
```