Word Order-Aware Text Processing: A Third Generation of Text as Data in Political Science

ANDREW HALTERMAN

Text has always been a key source of data for political scientists. In the past five to ten years, new techniques have automated some components of text analysis, allowing researcher to classify, categorize, and find meaning in large collections of text. I argue that text as data in political science has moved through two distinct approaches and is on the cusp of a third. The first generation consisted of simple text matching operations, where documents are searched for keywords or phrases were checked against hand-built dictionaries. The second generation introduced machine learning methods, operating on documents represented as "bags-of-words", where word order is discarded and documents are treated as counts of the words they use. The second generation has given us the staple techniques of text analysis in political science, including supervised document classification and topic modeling through latent Dirichlet allocation (Blei, Ng, and Jordan 2003) or the structural topic model (Roberts et al. 2013). The third generation will be characterized by word order-aware machine learning models, informed by research in natural language processing. These two techniques may perform better than the second generation on some existing tasks, but more importantly, this "linguistic turn" in political text analysis will enable wholly new methods of extracting meaning from text.

Each of these generations make assumptions about how to understand text, how to model language, and what kinds of models to apply to the text. In the following section, I describe how each approach arose to address shortcomings of previous approaches, what assumptions it is built on, and the classes of questions it is well suited to study, by highlighting example work in each. In the final section on third generation analysis, I discuss how several existing studies could be improved by extracting more relevant data from text than was possible at the time they were undertaken.

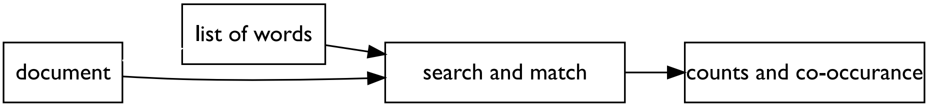
I would like to emphasize that while there is a temporal component to these generations, no strict boundary exists between their use and development. More importantly, the advent of new methods rarely supplants previous methods. New techniques occasionally outperform older techniques on existing tasks, but more often, they expand the range of possible tasks. Thus, text analysis in the future will continue to draw on all three approaches, using the techniques that are most suited to the task at hand, regardless of when they were developed.

An analogy to quantitative methods in causal inference is apt: more sophisticated causal models rarely improve the analysis of simple experiments or quasi-experiments. Their usefulness is instead in expanding the realm of observational data to which causal models can be applied. Increasing sophistication does not imply better inference in well-designed studies: it increases the range of data that can be used. Likewise, new text analysis methods generally expand the pool of analyzable data and answerable questions rather than simply improving previous analyses.[1]

First generation: matching terms and searching strings

The first generation of automated text analysis in social science was built to serve a limited task: to automate the laborious process of searching through text to find a set of pre-defined keywords or phrases or to count the words used in text. It arose in the early 1960s (Stone et al. 1962), but only took off after the late 1980s with the advent of widespread computing and the availability of digital text. Since then, the technique has been used whenever researchers have strong predictions about the usage of individual words and when words' presence or absence in text provides good evidence for or against a theory.

A typical workflow in this approach is quite straightforward. First, the researcher compiles a list of words or phrases (sometimes referred to as a "dictionary") to search for. The researcher than searches the text for these words or phrases, recording which documents they occur in, or potentially which documents certain terms or phrases co-occur in. These document counts are then used as evidence in advancing or testing a theory. Figure **??** summarizes this workflow.



{#fig:1st_gen}

These word lists can be created in several ways. Researchers can compile them manually, using *a priori* theoretical knowledge. For instance, the term "human rights" alongside a country's name in a news reports is expected to convey derogatory information about a country's human rights record (Nielsen 2013). Researchers can also draw on existing lists of words, for instance searching for place names to geolocate text (Nielsen 2013; Toft and

[1] There are of course exceptions. Using raw word counts to characterize corpora has been supplanted by topic models, and using the presence of keywords to classify documents has been replaced with document level machine classification.

Zhukov 2015; Douglass and Harkness 2018). Differences in what words partisans use in their speeches can be used to generate lists of terms that are used preferentially by liberals and conservatives; this list can then be used to classify news coverage (Gentzkow and Shapiro 2010). Finally, human coders can create lists of phrases with defined for political actors or events and assign them defined codes. A search through sentences to find these labeled phrases can produce a set of sentences matching these phrases and by filtering out sentences with two actor matches and an event phrase match, thereby a dataset of events (Schrodt, Davis, and Weddle 1994; Schrodt 2009). Tbl. 1 summarizes these tasks and the sources of the dictionaries that this method relies on.

TABLE 1 *Dictionary methods in political text analysis*

Task	Word Source	Citation
Geocode violent events using searches for place names	local gazetteers	(Toft and Zhukov 2015; Douglass and Harkness 2018)
Find and code events in text	hand-labeled text	(Schrodt, Davis, and Weddle 1994; Schrodt 2009)
Measure human rights abuses by searching for usage of the term "human right(s)" and country names in media	researcher decision	(Nielsen 2013, 726)
Measure US newspapers' ideological slant with word usage	terms in Congressional debate	(Gentzkow and Shapiro 2010)
Characterize the topics of Chinese blog posts	inductively from important topics	King, Pan, and Roberts (2013)

The advantages of the first generation dictionary approach are great and explain why this is still the first approach for many text analysis tasks. First, dictionary approaches are transparent. The presence or absence of a term in text is unproblematically and deterministically measured, and the list of search terms is easily inspected and debated. The process is also easy to implement. Search lists are easy to generate manually or with external data (the major exception being the very detailed lists required by automated event data coders) and searches are fast even over very large corpora. All of this stands in contrast to later machine learning methods, which often require much more data and effort to train, can be unstable across different runs, and can not always easily explain why a document was categories or summarized the way it was.

This approach comes with several major disadvantages, however. First, while it can easily locate very discrete terms ("arson" or "Senate", by searching for those words) it cannot be used to find more general concepts ("Snowden/surveillance/NSA") without a great deal of work and high false positive rates to define a comprehensive dictionary of terms. It also generally cannot take context into account: an attempt to find cabinet firings would struggle to distinguish between cabinet members being fired, members firing their subordinates, and cabinet members commenting on wild fires.

In terms of how it models documents and language, first generation approaches are largely model free. The simplest word search methods implicitly treat documents as bags-of-words, presence of a word is what matters, regardless of its location in the text. Slightly more sophisticated keyword searches require matches to occur within a set number of words from each other, relaxing the strict assumption of word order irrelevance. The most sophisticated word search methods can begin to approximate word order aware methods by carefully constructing phrases to search and making assumptions about the structure of certain kinds of text. Specifically, the automated event data systems KEDS and TABARI (Schrodt, Davis, and Weddle 1994; Schrodt 2009), exploited the rigidity of Reuters ledes and the regularity of English grammar to find the subject, noun phrase, and direct objects in news wire opening sentences, in order to recognize political events in text. Finally, modern event data systems (e.g. Petrarch, Schrodt, Beieler, and Idris (2014)), are a hybrid of first generation and later natural language processing-informed models, using NLP software to grammatically parse the sentence and then to check only the extracted noun and verb phrases against the keyword dictionaries.

The tools developed in the first stage of automated text analysis accomplished the tasks they were developed for: to find usage of words or phrases identified by the researcher or to count the occurrences of word in text. To be clear: for many tasks, the methods in this family of approaches are still the best to use.

Second generation: machine learning on bags of words

The second generation of text analysis grew out of clear limits on the first generation's usefulness to wider tasks. First among these is the lack of researchers' ability to use them to *explore* their corpus beyond rudimentary word frequency counts, rather than to approach them searching for pre-defined terms. The ability to explore a corpus depends on methods to summarize, cluster, or condense text and to discover what themes or topics it contains, potentially serendipitously, in a way that searches for keywords will rarely provide. The second shortcoming of first generations approaches comes from their reliance on researchers specifying the words are of interest. Political scientists often do not have good *a priori* theories about which words are important or will be used in which ways. We would prefer to learn which words are particularly salient for measuring certain concepts by operating at higher levels, such as the document, where we do have good intuitions or theoretical priors.

We can generate labels for documents by labelling them manually and then using an automated process to learn which words are predictive of our labels. We can also automatically learn clusters of correlated words, and then label these clusters with substantive knowledge, rather than specifying each word's topic importance directly. Both of these would mean researchers no longer need to approach their corpus with a defined dictionary of words in hand.

The second generation of text analysis began with the first efforts to model text using statistical tools, both in a supervised approach to predict labels for new text, and in an unsupervised way to cluster documents in a useful and interpretable way. This growth in modeling was made possible by an assumption that word order is largely irrelevant for many tasks, which works remarkably well for most document classification and clustering tasks. This approach, of applying models to bag-of-words documents, with or without labels, has become largely synonymous with text analysis in political science. In Wilkerson and Casas (2017)'s annual review of "Large-Scale Computerized Text Analysis in Political Science", the assumption, with only exceptions for event data and some work by computer scientists, is that researchers will begin by stemming, removing stopwords, and creating a document-term matrix.

Document representation: bag of words

The simplifying assumption that makes these machine learning methods possible is that the order of words in a document can be discarded without losing too much information in the text. Representing a single word on its own requires a large amount of information. Modeling the sequence of words on top of that results in a sequence that is likely to be utterly unique and vastly high-dimensional. By removing word order, each document can be represented as a vector the length of the language's total vocabulary, with elements representing the number of times a give word appears in a document.[2] By collapsing all words in a document to a single vector, a document can now be represented as a fixed length document of around 5,000 or 10,000 (the size of the vocabulary), as opposed to a variable length vector tens or hundreds of times longer, if word order is preserved. This assumption makes machine learning tractable and works remarkably well for many tasks.

Much of the methodological literature in the second generation concerns methods for preparing documents for analysis as bags of words, including stemming and lemmatization, stopword removal, removal of high- and low-frequency words, and the creation of n-grams or noun phrases. These decisions can sometimes be highly consequential (Denny and

[2] This vector is often scaled to downweight frequent words and boost unusual words using a "term frequency–inverse document frequency" approach.

Spirling 2018).

Numerical summaries of documents

Virtually all models in the second generation of automated text analysis take in a bag-of-words representation of a document and produce a useful lower dimensional output or summary of the document.[3] Specifically, the document can be summarized as

1. a binary score, making the task a document classification task
2. a continuous score, as is the case in ideological scaling, IRT, and some sentiment models
3. a low dimensional vector representation, the most common of which are topic models

Each of these model outputs are useful for different tasks in text analysis.

Figure **??** shows the conceptual workflow of second generation methods. Documents are converted to a bag-of-words (or n-grams) representation, and different functions are available to map the document to a binary category, a continuous score, or some low dimensional representation. Tbl. 2 shows examples of these tasks. Formally, if each word w_i in a document of length M is a unique "one-hot" vector (with all elements 0 except a single 1 value) of length V, where V are the unique words in the vocabulary, a document D can be represented as $D = \sum_i^M w_i$, so D will also be a vector of length V, but no longer one-hot. Different functions are available that map this document vector to lower dimensional space:

$$
\begin{aligned}
f &: D \to \{0, 1\} \\
f &: D \to \mathbb{R} \\
f &: D \to \mathbb{R}^k
\end{aligned}
$$

[3] I use bag-of-words synonymously with the more general but clunkier term "bag-of-n-grams". N-grams include additional terms in the vocabulary for for neighboring words, allowing potentially important meaning to be preserved, as in the case of "social_security", compared to {"social", "security"}. This clearly has advantages and can preserve some important word order information. The cost comes from exponentially increasing the number of potential words in the vocabulary or decisions about how to prune the vocabulary. Spirling (2012) provides a rare example of a model that does not rely on the bag-of-words assumption. Specifically, he represents documents as the set of all short sequences of characters in a document. Using kernel PCA, the similarity between two documents can be obtained in a way that preserves much of the word order information in the two documents.

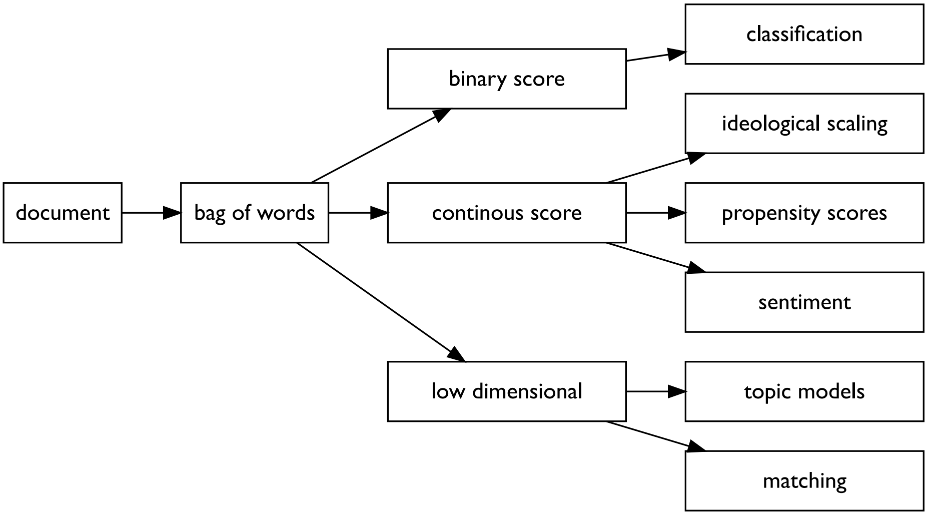


Figure 1. Second Generation Text Analysis: Document Representation and Tasks

Binary scores place a document into a category using supervised learning, with the output being either a hard assignment to a category or a probability of membership in a class. Documents can also be classified into one of k exclusive categories, but this is a generalization of the binary case. Continuous scores are generally used in political science for scaling text along an ideological dimension. Documents can also be placed on a sentiment scale, though this application is rare in political science. A continuous score could also represent a propensity score, if researchers were to apply propensity score matching to documents. Both supervised and unsupervised methods can be used here. Finally, low-dimensional representations represent documents with a dense vector in low dimensional (e.g. 5-50) space. Low dimensional representations include topic models, the most common form of text analysis in political science. The outputs of topic models, the word composition of topics and document proportions of topics, are generally the objects of interest. In some cases (see Tbl. 2), the topics can be used for document matching instead.

TABLE 2 *Examples of second generation text analysis*

Description	Representation	Model	Citation
Classify regime type using government and NGO reports	binary score	SVM	Minhas, Ulfelder, and Ward (2015)
Understand which issues Congress debates	low dimensional	LDA	Quinn et al. (2010)

Description	Representation	Model	Citation
Estimating the ideological position of political parties from manifesto text	continuous score	EM Poisson of words	Slapin and Proksch (2008)
Predicting legislators' votes from bill text	low dimensional	ideal point LDA	Gerrish and Blei (2011)
Learning conservative and liberal terms from a classification model on political speeches	binary (feature weights of interest)	SVM	Diermeier et al. (2012)
Estimate topics and correlation within a corpus of Islamic cleric's text	low dimensional	structural topic model (extension of LDA)	Nielsen (2017)
Estimate document ideology from coded sentences	continuous	IRT	Benoit et al. (2016)
(Proportion of blogs with) positive or negative sentiment toward presidential candidates	continuous	SVM	Hopkins and King (2010)
Classifying news articles for signals of impending mass killing	binary	SVM	Halterman, Ulfelder, and Valentino (2016)
Do legislators engage in credit claiming or issue taking?	low dimensional	(hierarchial) LDA	Grimmer (2010)
Can matching on topics recover causal effects and be interpretable?	low dimensional	STM	Roberts, Stewart, and Nielsen (2018)
Can propensity scores be used for text matching?	continuous score	logistic regression on topics	Mozer et al. (2018)

What these tasks have in common is that they operate at the level of a document, as opposed to smaller pieces of text, and represent documents as a bag of words. This bag of words representation is then mapped onto a lower-dimensional output. Crucially, where first generation approaches required researchers to have strong prior assumptions about the importance of individual words, these methods allow the significance of words to be learned from their co-occurrence in documents or their ability to predict a document label. The advent of second generation models, machine learning on bags-of-words, is responsible for the recent rise in automated text analysis. For most document classification, ideological scaling, and topic detection tasks, these methods will not be superseded by other methods.

Third generation: word order-aware methods for information extraction (the linguistic turn)

The emerging third generation of text analysis addresses gaps in the dominant second generation approach. The primary gap in second generation methods is in the form of information that can be extracted from the document, namely, whether information about the document as a whole is of primary interest, or whether specific subsets or components of the document are of interest. Second generation models are useful when the document itself is of interest, answering questions such as how people speak or write, what the topical contents of a corpus are, or how documents can be classified or summarized. These document-level methods are not well suited to extracting specific information from text: who did what to whom or what relationships exist between entities. In a more abstract epistemological sense, second generation methods are best suited for studying discourse, speech, and belief inherent in the text; third generation methods treat text as a vehicle for conveying factual information about the world.

The nascent third generation of text analysis uses improvements in machine learning and natural language processing to model text more accurately, accounting for word order and using the syntactical information of sentences. Its real advantage is allowing new tasks and approaches, primarily in the realm of information extraction. The second generation of text analysis generally operates at the level of the document and produces as output either a single score (classification) or a low dimensional representation (clustering, topic modeling). Information extraction, in contrast, can operate at the sentence level to extract specific pieces of data in the text, including events that occur, relationships between political actors, their locations, and specific reported attributes of actors. These techniques will allow researcher to move beyond the discourse-oriented approach of generation two and begin to treat documents as sources of facts to be mined.

Representing documents

The ability to extract information from documents as opposed to converting documents to whole-document summaries, rests on new ability to represent documents and to model the sequential nature of text. While much of this improvement comes from new models, specifically deep neural networks, the more important breakthroughs have come in better numerical representations of words and sentences, and the ability to share this information across different applications. Third generation text analysis can use a wealth of syntactic information to extract meaning from a sentence and to overcome the extreme sparsity of representing each word by its number in the vocabulary.

In the past five years, the state of the art in natural language processing has shifted from representing words as one-hot vectors to an approach that represents words as dense, low-dimensional vectors called word embeddings (Mikolov, Sutskever, et al. 2013; Levy and Goldberg 2014)[4]. Word embeddings are a method of learning a low-dimensional representation of words from text based on local co-occurrence with other words. The "distributional hypothesis" in linguistics holds that "a word is characterized by the company it keeps" (Harris 1954). Word embeddings represent words as much shorter vectors (usually 50–300) that are dense, meaning each dimension has a continuous value. The embeddings can be learned as a hidden layer in a shallow neural net that tries to predict a word based on its neighbors (Mikolov, Sutskever, et al. 2013) or as a matrix factorization on local word co-occurrence (Levy and Goldberg 2014). These dimensions, like the "topics" in topic models, have no inherent meaning, but are learned automatically from the text and may in some situations have an interpretation. Once trained on a larger collection of text, these pre-trained word embeddings can be transferred to new tasks and domains. By using outside information on how words are used, models that employ word embeddings can be much more powerful with much less data.

Approaching documents from an NLP perspective also has the advantage of providing a range of grammatical information about a sentence. Documents can be modeled syntactically, as sequences of part-of-speech labels or as dependency trees that encode the syntactic relationship between words. This approach is especially useful for tasks that involve recognizing particular pieces of the sentence, including named entities or noun phrases, or for linking parts of a sentence together, such as linking actors and their actions or events and their locations.

Information extraction

Where second generation text analysis represents documents as bags of words or bags of n-grams, third generation text analysis has a wealth of document representations. The most similar to bag-of-words is a simple mean of the document's word vectors. Here, the word order of the document is discarded, but the model uses pre-trained word embeddings trained

[4] **Question**: Do I need a figure here to explain word embeddings? And a poli sci example using custom trained embeddings, say on cables?

on local word co-occurrence to improve the representation of the document. These averaged word vectors are useful for document clustering or simple document classification, in situations similar to when an SVM on a bag-of-words would be appropriate.

The other document representations are more sophisticated and enable much more interesting information extraction tasks. A second approach is to simply model the document as a sequence of words. This representation strategy runs into the same issues that motivated the bag-of-words assumption in the second generation, namely that a document's representation will be $V \times N$, where V is the total size of the vocabulary and N is the document length. This representation has become somewhat more tractable[5] with greater availability of text and models that can share weights across a sequence of inputs (specifically, recurrent neural networks such as LSTM).[6] A small increase in sophistication, using a sequence of word embeddings, results in a drastic increase in accuracy. Here, each word is represented as an embedding, usually pre-trained[7] and the document is modeled as a sequence of these embeddings. A recurrent or convolutional neural network can then be applied to the sequence of embeddings to predict a label for the document, the part of speech tags for each word in the document, the grammatical or substantive relationship between two words in a document, or recognize names in the text.[8]

[5] The size of the document matrix is not an issue for memory, because the matrices are extremely sparse and can be modeled as such, but instead an issue for estimating the position-specific effect for each word.

[6] "long short-term memory" (LSTM) models are an attempt to overcome the "vanishing" or "exploding" gradient problem that arises when parameter updates need to be backpropogated through more than a handful of steps or layers (Hochreiter and Schmidhuber 1997). LSTMs partially over this problem by allowing the neural network to learn when to add information to its memory cell (a hidden vector) and when to "forget" from the memory cell.

[7] Most models use pre-trained word embeddings from a large corpus like the Google News corpus or Wikipedia. Some use pre-trained vectors but allow them to be updated during model fitting. Both of these approaches are examples of transfer learning, where a model trained on one task can be transferred to a different but related task and thereby the reduce the amount of training data needed in the second task. The final approach is to initially represent each word as a randomly drawn vector and to backpropagate to this layer, training the word embeddings from scratch along with the model. This approach requires a large amount of task-specific labeled data and may result in overfitting.

[8] The sequence of embeddings can also be a sequence of *character* as opposed to *word* embeddings. This approach works "unreasonably" well (Karpathy 2015) across a wide number of domains, including in event data generation from text (Beieler 2016). Convolutional neural networks (CNNs) do not have the same appealing theoretical properties of recurrent neural networks (RNNs), in that RNNs explicitly incorporate sequencing information and CNNs were designed in the image recognition case to ignore input within the input. Recent

Text can also be modeled as a sequence of linguistic information generated by natural language processing software (see Tbl. 3). This information can include part-of-speech tags, named entity recognition tags, and the dependency parse of the sentence. Dependency parses include the grammatical relationships between words in a sentence, marking, for instance, which nouns are direct objects of a verb, which adjectives modify a noun, and so on. Knowing whether a word is a noun or verb would solve our cabinet member (wild)fire example from above: a political scientist studying personnel turnover is interested in "fire" the verb, not "fire" the noun. This information can be combined with word embeddings as additional features for a model, though some argue that it maybe better to have the model learn this grammatical information itself.[9]

TABLE 3 *Word annotations and representations in natural language processing*

Text Feature	Description
part-of-speech	the part of speech (noun, verb, preposition, adverb, etc.)
dependency label	the syntactic relationship between words (e.g. word 3 is an adjective modifying word 4)
named entity label	the type of entity a token is (if any), including person, location, organization, etc.
word embedding	a low dimensional vector (compared to the vocabulary size) learned from word usage, such that semantically similar words are close in this space
character embedding	similar to word embeddings, but at the character level. These learn useful subword information, such as the role of "-ing" or "pre-" in words.
lemma	a base form of a word (e.g. the lemma form of "said" is "say")
word shape	the length, punctuation, and capitalization patterns in a word. This is especially useful for recognizing named entities in languages that use capitalization.

Figure 2 displays the new tasks that are possible in the word order aware paradigm and

empirical work, however, has shown that stacking many convolutional layers, with tricks to increase the "receptive field" of the network while avoiding gradient problems, performs just as well as CNNs on sequence modeling tasks (Bai, Kolter, and Koltun 2018). CNNs over the tremendous advantage of being easily parallelized in training over convolutions, while RNNs must model the entire sequence at once.

[9] See Collobert et al. (2011) for an early example of where learning to optimize several outputs at once improves performance on each task separately, as the system learns a better internal representation of a sentences' syntax and meaning.

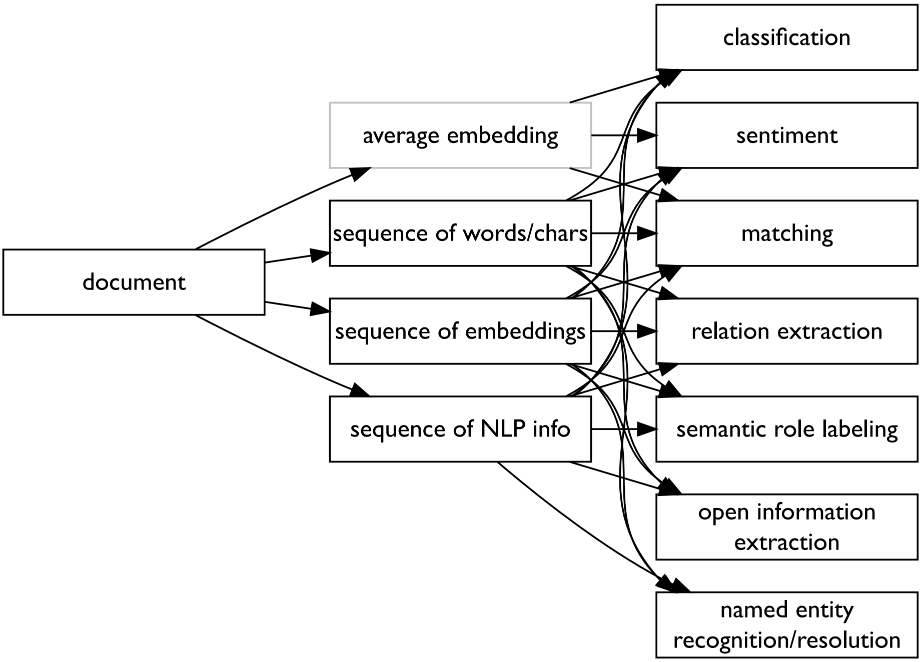


Figure 2. Word Order-Aware Document Representations and Information Extraction

how they can be used to extract information from text. The tasks that this new information makes possible consist of, in increasing sophistication:

1. named entity recognition, resolution, and coreference
2. binary relation detection
3. semantic role labeling
4. open information extraction

Named entity recognition (NER) is the ability of a language model to recognize proper names in text (people, organizations, locations, etc.), but also other kinds of entity-like text (dates, currency, references to named events), including custom, task-specific entities such as the names of bills and laws, named battles, agency names, drugs, weapons, or military units. Simply using named entity information on its own will likely be more useful in exploratory research in political science, making it easy to find, for example, which American diplomats were most involved in discussing the Vietnam war from declassified State Department cables or as a case selection technique, finding the place names most often mentioned in legislative debate on the effects of globalization. See Tbl. 4 for more potential research applications.

Named entity recognition is useful for theory testing as well. James Dunham is extracting the names of organized interests from transcripts of public hearings to measure the effect of unpaid lobbying by advocacy groups, behavior that does not appear in formal lobbying filings. Named entity information could also be used in a similar way to the way topic models measure the attention paid to issues by bureaucracies in their written reports, by measuring changes in the places in the world receiving the greatest attention from State Department diplomats.

Generally, to be useful, named entity recognition requires an additional linking step, resolving different references to the same entity into a common form. For instance, "President Clinton" and "Bill Clinton" refer to the same person, but "Secretary Clinton" does not. This step usually requires references to an external knowledge base such as Wikipedia, can also be learned from information provided directly in the text, such as "Mrs. Clinton, the US Secretary of State, …". This latter task is a version of entity coreference resolution, which can link "Clinton" in a document to an earlier mention of "Secretary of State Hillary Clinton", or in the case of pronoun coreference, link "she" to the appropriate name. Most named entity systems will not be useful without these linking steps if the identities of the named entities are important.

TABLE 4 *Named entity recognition in political science*

Research Question	Operationalization	Implementation	Why 3rd Gen?
What does the executive pay attention to in foreign policy? Is it reactive or proactive? Do human rights violations attact attention? (Similar to Quinn et al. (2010)).	Extract people and locations from the declassified President's Daily Brief (1961-1977) and resolve to country. Compare with newspaper coverage and to known events.	standard entity labels, perhaps improved to match idiosyncratic text	Unlikely to have list of all locations, leaders, and politicians. Better to categorize after the fact.
Does the balance of civil-military relations change in the lead-up to a coup?	Compare mentions of civilian and military leaders in declassified State Department cables.	Existing entity labels, tweaked to work on cables. May need text-based role resolution to code actors' affiliation	Unlikely to have lists of all civilian and military personnel

Research Question	Operationalization	Implementation	Why 3rd Gen?
(Research design phase) Who are the major actors operating in a particular context? Who should a researcher investigate more fully to ensure they have understood the full sample of political actors?	Examine the set of named entities extracted from text relevant to the case (archival data, previous academic publications, news reports). Examine entity prevalence by different topics from a topic model	standard named entity recognition software, with topic models	Because the purpose is to find actors the reseacher may not have known about before, pre-made lists of entities are not feasible.

(Binary) relation extraction is the process of linking named entities according to a specific relationship (see e.g. Doddington et al. 2004; Li and Ji 2014). For instance, a binary relation extractor could look for a person's membership in a group, which can be very useful when researchers are interested in studying the behavior or rebel groups or legislative committees, rather than the behavior of individual rebel leaders or committee members. For well-known people, this membership is often reported in a knowledge base such as Wikipedia, but even this may require natural language processing to extract (e.g. "Curren De Mille Price, Jr. (born December 16, 1950, in Los Angeles, California) is an American politician of the Democratic Party, currently serving as a Los Angeles city council member for District 9."). Other binary relation types that are potentially useful in political science are locations of birth, citizenship in a country, geographic hierarchy (a village is in a particular district), or party membership.

Semantic role labeling (SRL) is the most sophisticated of these standard natural language processing tasks. It consists of recognizing a defined "frame" in a piece of text and filling in the slots defined for each relationship (see, e.g., the FrameNet project, Baker, Fillmore, and Lowe (1998), and PropBank, Kingsbury and Palmer (2002)). For instance, and SRL system could be designed to extract events or "frames" of the form "[group X] captured [location Y]". A well functioning system would then need to recognize that all of the following sentences consist of the same "capture territory" frame with the same actor and location:

- ISIS captured Palmyra.
- ISIS has captured the town of Palmyra.
- ISIS has overrun and gained control of Palmyra.
- Palmyra has fallen to ISIS.
- Government forces have retreated from Palmyra, leaving it to fall to ISIS.

Semantic role labeling is thus a general form of traditional event data in political science, approaching it from a computational linguistics angle, beginning with the grammatical structure of the sentence, in contrast to political science event data, which has traditionally relied more on pattern matching and hard-coded rules. The promise of NLP-informed event extraction is in increasing accuracy and greatly reducing the word required to create a dataset, when compared to previous dictionary-based methods. The primary difference between semantic role labeling and binary relation extraction is that SRL needs to have a more sophisticated treatment of verbs. The applications of semantic role labeling in political science research are great. In addition to measuring changes in territorial control, which is relevant for my research and others (e.g. Raleigh et al. 2010; Tao et al. 2016), SRL can be used to extract events that are currently being extracted by hand (see Tbl. 5).

TABLE 5 *Semantic Role Labeling in Political Science*

Research Question	Operationalization	Author
To what extent is civilian death in civil war the product of conventional military forces and operations?	[armed group X] launched an offensive in [location Y]	Andrew Halterman
What is the effect of military diplomacy on alliances?	[navy X] made a port call in [location Y]	Jennifer Spindel
How can the U.S. encourage cooperative behavior from its military partners?	[military X] conducted joint exercises with [military Y]	Rachel Tecott
Do areas in Europe receiving more migrants experience greater amounts of right wing violence?	[European right wing group X] conducted an attack of [type Y] against [target Z]	Jill Irvine
What are the causes and effects of collective political action?	a protest with [tactics X] occurred in [location Y]	Chenoweth, SCAD, Hanna, Miura, and others

A fourth task that word order-aware approaches make possible is "open information extraction". Open information extraction extracts subject-verb-object triples from text but does not attempt to categorize them in a pre-defined way, as semantic role labeling and relation detection do. Take the following sentence as an example.

> Last week, ISIS launched an offensive near Dumayr Airbase and subsequently captured Badia Cement Factory from government troops, even kidnapping some 300 Syrian workers from the site.

An open information extraction system would extract several subject-verb-object triples from this sentence, including ("ISIS", "kidnapping", "some 300 Syrian workers"). Open information extraction tools use rules and parts of speech and dependency labels for extracting the triplets (Banko et al. 2007; Mintz et al. 2009 Chambers and Jurafsky (2011); Ritter et al. 2012). The advantage of this approach is that it does not require specific rules or models to extract only specific types of interactions, as semantic role labeling does. Once extracted, the components can be labeled with dictionary methods or can be clustered. O'Connor, Stewart, and Smith (2013) employ a related approach in which they search for state actors using dictionary methods, extract the verb phrase in between them, and run a topic model on the extracted verb phrase. (Aside: I've been thinking over the past few days about a better way of modeling these triplets. I've been thinking about a custom correlated topic model that models words within triplets within documents, and allows for correlation between the topics occurring in the three slots. This would allow actor- and action-specific topics to emerge and the correlations between subject, object, and action topics to be correlated. There's some interesting math that would be involved and it would be an interesting bridge between bag-of-words topic models and word order aware NLP methods. I would be very interested in hearing your thoughts on this idea during the colloquium.)

This section has highlighted only four major information extraction for being the most important and applicable to political science. Information extraction includes many more tasks than those mentioned here (for instance, the task of recognizing and resolving dates in text), or are much more difficult than my brief references would imply (entity co-reference is far from a solved problem). Jurafsky and Martin's *Speech and Language Processing*, 3rd Ed (draft)[10] offers an excellent overview of information extraction and NLP more generally.

Comparing second and third generation approaches

Before moving onto a more detailed discussion of the possible applications of word order-aware third generation techniques, it is useful to pause and compare the major differences between second and third generation methods. Tbl. **??** summarizes the general differences between the two methods.

TABLE 6 *Comparison of Second and Third Generations*

.	Second Generation	Third Generation
Word Order?	no	yes
Language Model	generative or none	empirical and opaque
Output	document summary	extracted information
Measurement Target	discourse, ideas, concepts	information in text

[10] Available at https://web.stanford.edu/~jurafsky/slp3/

The most apparent difference between the second and third generations is in how they model language. The second generation assumes a bag-of-words model in which word order in the document is discarded. At the same time, many second generation models also have very sophisticated and explicit statistical models of how documents are generated, even if they are unrealistic as an account of how humans write documents. For instance, the structural topic model (Roberts et al. 2013) models how each word in a document is a draw from a topic's probability distribution over words, where the topic's prevalence and composition is affected by document-level covariates. Some second generation techniques, specifically classification methods such as SVM or naïve Bayes, have no generative model for documents. Third generation approaches preserve word order information, making them closer matches to the way language is used, but generally do so at the cost of having explicit or interpretable formal models of how text is generated. Models of text in this approach rely on empirical examinations of how many documents are written to construct relatively opaque statistical models.

Another key difference is in what the ultimate target of measurement is. In second generation analysis, the output of most models is a numerical summary of a document, and the object of interest is itself the document (it is literally text analysis). In the third generation, the text is instead often treated as a repository of facts and information to be extracted. In the information's application to substantive questions, their origin in text is largely irrelevant. Again, these differences highlight the complementarity of these two approaches. Second generation methods are useful for inference about documents and their substantive content, while third generation approaches are generally more useful for extracting information from them.

How third generation approaches could improve existing studies

While most of the applications of third generation methods, including those in my dissertation, are to producing new data that was previously infeasible to make, some previous studies may be improved with advances in third generation methods. The bag-of-words assumption, necessitated by the methods previously available, limits the kinds of information that can be extracted from text with previous methods, meaning that some previous studies may be improved by a more precise textual measurement strategy.

Perceptions of controversial figures in news text. In their survey of text analysis methods for comparative politics, Lucas et al. (2015) present an example of multilingual text analysis to study varying perceptions of Edward Snowden in Chinese and Arabic-language social media. The authors use a topic model to measure the topics or themes mentioned alongside the name "Snowden", and find that a topic concerned with American "attacks" is much more prevalent in Chinese than in Arabic, while Arabic tweets discuss asylum and other factual matters from the case. As a technique for measuring public perception of the United States, the NSA, and spying, a topic model is completely appropriate. An alternative research question might be interested in the way that Snowden himself is described across media sources. In order to answer that question, topics would need to be generated from the sentences or clauses that directly address Snowden, which becomes impossible when a document is converted to a bag-of-words representation. Instead, a pre-processing step could extract adjectival or appositive phrases attached to mentions of Snowden and fit a topic model on just those phrases in order to understand regional variance in how Snowden was treated.

The same approach, of topics linked more closely to entities, could be used to study perceptions of other political actors in text, especially actors frequently mentioned alongside unpleasant events. For instance, news coverage of police will likely be correlated with crime and violence: studying, for instance, changes in news coverage of police violence would depend on stronger links between topics and police themselves. Preserving word order and grammatically linking actors with their descriptions in text provides a way to more precisely use text to measure certain concepts.

Jihadi clerics. In his recent book, *Deadly Clerics*, Nielsen (2017) studies the paths to radicalization on the part of Muslim clerics. One set of data he draws on is a sample of 200 cleric biographies. He uses several techniques to explore the contents of these biographies. He is interested in characterizing both the biographies themselves (what do clerics emphasize about their training? how are they constructed?), as well as extracting specific pieces of information from them (how many teachers, how many locations, what degrees they hold). Notably, these latter pieces of information were gathered either manually, in the case of teachers, or through a keyword search of country names to approximately count the distinct locations in a piece of text (pg. 100-101) and educational terms to measure the level of education. Some of this information could be better extracted using named entity recognition and relation extraction, though the increased fidelity and the lower marginal cost per document may not be worth the up front effort to train or validate a machine learning model.

A later chapter is concerned with recognizing jihadi writing. The task of classifying writing as jihadi or not is unlikely to be drastically improved with word order-aware methods, though some documents misclassified with the bag of words naive Bayes model in the book may be correctly labeled with a more sophisticated model, as Nielsen points out (2017, 125). Where word order-aware methods could be useful is in understanding the parts of a document that are most responsible for it being classified a certain way. Recent "attention" models in natural language processing can generate per-character contributions to document classification, which can be used to recover phrases or combinations of words that are responsible for large shifts in document classification. Combining this information with close reading of documents can produce new insights into terms or phrases that jihadi clerics use that may not have been expected.

The causes of censorship in China. King, Pan, and Roberts (2013) have one of the best known applications of text analysis in political science. They collect posts made by Chinese bloggers and record whether the post is eventually censored. They find that posts that contain words related to events that have a "collective action potential" are much more likely to be censored than posts that do not, even when those posts criticize the government. The keyword-based methods they employ are completely transparent (they report the terms in their appendix) and were adequate to the task they pursued.

New text methods developed in the five years since the article's publication could offer new avenues to learning from the data. First, the task could be reformulated as a supervised learning task. What terms or phrases best predict censorship? Rather than needed to specify the potential censorship-causing terms before the analysis, a model could learn which terms are predictive. This task is easily done in a bag-of-words framework (as Diermeier et al. (2012) did on Congressional speeches), but could also been done in a way that preserves word order. Recent work in neural network-based classification (Yang et al. 2016) has focused on "attention" mechanisms that place greater weight on certain contiguous words or characters text in making the document classification decision. The high-attention parts of a document can be extracted as highly predictive phrases. Other work from MIT (Lei, Barzilay, and Jaakkola 2016) formulates a classification model explicitly to return short "rationale" pieces of text to justify a document's classification. These methods offer ways of learning the phrases that are mostly likely to result in censorship.

Another NLP-informed approach would build on the finding in King, Pan, and Roberts (2013) to more precisely test the collective action mechanism they identify. A blogger discussing a topic with high "collective action potential" is not necessarily advocating collective action. Likewise, a blogger may be encouraging collective action or protest around a mundane topic that does not rise to the level of discussion the authors use to determine a potential collective action event. Measuring whether a blog post is calling for collective action is likely to require a more nuanced model of human language than the bag-of-words representation is capable of providing. To further test their theory that collective action, not criticism, is the basis of censorship, a researcher could hand-label documents that indeed call for collective action. If a machine learning classifier could replicate the labels, it could be used for a more direct test of the authors' preferred theory.

Research agenda

The tremendous strides that computer scientists and computational linguists have made in analyzing text cannot be unproblematically applied to text in political science. Work is needed to modify their techniques to the problems and unique text in political science, to develop new models for political science text, and to place these techniques in a context where measurement and inference are more valuable. A large literature has emerged in the past five years to provide guidance to researchers using bag-of-words models for text, including on preprocessing documents, selecting models to use, and how to draw inference from text. This literature will need to be extended to address new word order aware techniques, potentially along the lines I describe above.

In addition to these implementation questions, several broader questions about information extraction and text analysis with machine learning need to be addressed. First, researchers have reached some consensus about the appropriate techniques for preprocessing documents for second generation analysis, including stemming, (in)frequent word removal, and lowercasing (Grimmer and Stewart 2013; Denny and Spirling 2016).[11] How should documents be represented in word order-aware ways? Standard approaches in computer science use pre-trained word or character embeddings to greatly reduce the labeled data required to reach good accuracy. Should word or character embeddings be used for political science applications? How concerned should researchers be about the well-known biases in pre-trained embeddings being propagated through to their findings? The standard off-the-shelf embeddings, word2vec (Mikolov, Chen, et al. 2013; Mikolov, Sutskever, et al. 2013) and GloVe (Pennington, Socher, and Manning 2014), contain clear sterotypes and bias, including strong associations between gender and stereotypically gendered occupations and the association between typically African-American names and "unpleasant" terms (Bolukbasi et al. 2016; Caliskan, Bryson, and Narayanan 2017).

Second, researchers need new techniques to model and adjust for reporting bias in text. When the corpus of text is itself the object of inquiry, as in the case in many studies using topic models, exclusion from the corpus is not an issue or not defined. In third generation approaches, where the objective is to extract information from text for use in future analysis, the problem of under-reporting becomes major. While the best solution is to use text corpora that are likely to accurately report the information of interest, not all text collections will. Better techniques, building on existing approaches (Hendrix and Salehyan 2015; Weidmann 2016; Bagozzi et al. 2018), will be needed to appropriately caveat findings that build on information extracted from text.

Because third generation information extraction techniques are statistical, they will contain errors. Entities or relationships will be missed by a model, while some false positives will be returned. Fortunately, because most information extraction is supervised, we can estimate the error rates of information extraction techniques. Guidance is needed on how to propagate information extraction error through to uncertainty in our final estimates.

Most third generation techniques are supervised machine learning models. In some cases, such as named entity recognition, researchers in political science will be able to use pre-

[11] update with new PA.

trained models. In other cases, such as document classification or domain-specific named entity recognition, researchers will need to label their own data and fit their own model. Methodologists need to provide guidance on the best steps to take in doing so, methods for determining marginal increases in accuracy with more data, and when to stop collecting data. Techniques in third generation text analysis will themselves speed this process by using machine learning models to determine the optimal documents to have humans annotate (Biessmann and Schmidt 2018).

Next, political scientists will need to independently investigate the performance of standard computer science models on real world applications in political science. Many training and testing datasets in computer science are unnaturally neat or contrived, which risks creating overconfidence on the part of their creators and users when applying them to messy text from another domain. State-of-the-art models from computer science may be overfit to the particular datasets they use or may provide only very marginal improvements with much greater computational costs, meaning that applied researchers in political science should perhaps use simpler models instead. Political methodologists will need to provide guidance on the appropriate models to use.

Finally, mixed method methodologists can investigate the usefulness of word order aware methods in early stage research or research design. Information extraction is optimized for learning about previously unknown entities, relationships, and events in text. Researchers can use these techniques on archival data, previously published academic studies, and on contemporaneous news text to ensure they understand the full range of actors involved in a case. They can use document geoparsing techniques (Halterman 2018) to geographically stratify archival documents for sampling and reading. Researchers can also use grammatically-informed topic models to understand the range of behavior and tactics of different actors reported in their text to build typologies and suggest hypotheses for future research.

Conclusion

This paper introduced a new framework for viewing text analysis in political science. Rather than dividing text analysis methods into supervised and unsupervised, as is often done, I argue instead for a division into three approaches, based on how each one models language and what information it returns about a document. Dictionary methods relying on manually constructed word lists arose first, followed by machine learning on bag-of-words. Now, new techniques are emerging, building on advances in natural language processing that allow researchers to account for word order. The benefits of these new techniques will be in extracting new kinds of information from text.

References

Bagozzi, Benjamin E, Patrick T Brandt, John R Freeman, Jennifer S Holmes, Alisha Kim, Agustin Palao Mendizabal, and Carly Potz-Nielsen. 2018. “The Prevalence and Severity of Underreporting Bias in Machine-and Human-Coded Data.” *Political Science Research and Methods*. Cambridge University Press, 1–9.

Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. 2018. “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.” *arXiv Preprint arXiv:1803.01271*.

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.

Banko, Michele, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. “Open Information Extraction from the Web.” In *IJCAI*, 7:2670–6.

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

Benoit, Kenneth, Drew Conway, Benjamin E Lauderdale, Michael Laver, and Slava Mikhaylov. 2016. “Crowd-Sourced Text Analysis: Reproducible and Agile Production of Political Data.” *American Political Science Review* 110 (2). Cambridge University Press: 278–95.

Biessmann, Felix, and Philipp Schmidt. 2018. “Speeding up the Manifesto Project: Active Learning Strategies for Efficient Automated Political Annotations.” *Manifesto Corpus Conference*.

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

Bolukbasi, Tolga, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. “Man Is to Computer Programmer as Woman Is to Homemaker? Debiasing Word Embeddings.” In *Advances in Neural Information Processing Systems*, 4349–57.

Caliskan, Aylin, Joanna J Bryson, and Arvind Narayanan. 2017. “Semantics Derived Automatically from Language Corpora Contain Human-Like Biases.” *Science* 356 (6334). American Association for the Advancement of Science: 183–86.

Chambers, Nathanael, and Dan Jurafsky. 2011. “Template-Based Information Extraction Without the Templates.” In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 976–86. Association for Computational Linguistics.

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. "Natural Language Processing (Almost) from Scratch." *Journal of Machine Learning Research* 12 (Aug): 2493–2537.

Denny, Matthew J, and Arthur Spirling. 2018. "Text Preprocessing for Unsupervised Learning: Why It Matters, When It Misleads, and What to Do About It." *Political Analysis*. Cambridge University Press, 1–22.

Denny, Matthew James, and Arthur Spirling. 2016. "Assessing the Consequences of Text Preprocessing Decisions." *SSRN*.

Diermeier, Daniel, Jean-François Godbout, Bei Yu, and Stefan Kaufmann. 2012. "Language and Ideology in Congress." *British Journal of Political Science* 42 (1). Cambridge University Press: 31–55.

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

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*.

Gentzkow, Matthew, and Jesse M Shapiro. 2010. "What Drives Media Slant? Evidence from Us Daily Newspapers." *Econometrica* 78 (1). Wiley Online Library: 35–71.

Gerrish, Sean, and David M Blei. 2011. "Predicting Legislative Roll Calls from Text." In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 489–96.

Grimmer, Justin. 2010. "A Bayesian Hierarchical Topic Model for Political Texts: Measuring Expressed Agendas in Senate Press Releases." *Political Analysis* 18 (1). Oxford University Press: 1–35.

Grimmer, Justin, and Brandon M Stewart. 2013. "Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts." *Political Analysis* 21 (3). Oxford University Press: 267–97.

Halterman, Andrew. 2018. "Linking Events and Locations in Political Text." *Draft Working Paper*.

Halterman, Andrew, Jay Ulfelder, and Benjamin A Valentino. 2016. "Mining News Reports for Predictive Signals of New State-Led Mass Killings." In *Paper Prepared for the ISA Global Trends Working Group, 57th Annual International Studies Association Convention in Atlanta,* *Georgia (March 16, 2016).*

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

Hendrix, Cullen S, and Idean Salehyan. 2015. "No News Is Good News: Mark and Recapture for Event Data When Reporting Probabilities Are Less Than One." *International Interactions* 41 (2). Taylor & Francis: 392–406.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9 (8). MIT Press: 1735–80.

Hopkins, Daniel J, and Gary King. 2010. "A Method of Automated Nonparametric Content Analysis for Social Science." *American Journal of Political Science* 54 (1). Wiley Online Library: 229–47.

Karpathy, Andrej. 2015. ""The Unreasonable Effectiveness of Recurrent Neural Networks."" *Karpathy.github.io/ (Blog)*, May.

King, Gary, Jennifer Pan, and Margaret E Roberts. 2013. "How Censorship in China Allows Government Criticism but Silences Collective Expression." *American Political Science Review* 107 (2). Cambridge University Press: 326–43.

Kingsbury, Paul, and Martha Palmer. 2002. "From Treebank to Propbank." In *LREC*, 1989–93.

Lei, Tao, Regina Barzilay, and Tommi Jaakkola. 2016. "Rationalizing Neural Predictions." *arXiv Preprint arXiv:1606.04155.*

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.

Lucas, Christopher, Richard A Nielsen, Margaret E Roberts, Brandon M Stewart, Alex Storer, and Dustin Tingley. 2015. "Computer-Assisted Text Analysis for Comparative Politics." *Political Analysis*, mpu019.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Efficient Estimation of Word Representations in Vector Space." *arXiv Preprint arXiv:1301.3781.*

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.

Minhas, Shahryar, Jay Ulfelder, and Michael D Ward. 2015. "Mining Texts to Efficiently

Generate Global Data on Political Regime Types." *Research & Politics* 2 (3).

Mintz, Mike, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. "Distant Supervision for Relation Extraction Without Labeled Data." In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Acl and the 4th International Joint Conference on Natural Language Processing of the Afnlp: Volume 2-Volume 2*, 1003–11. Association for Computational Linguistics.

Mozer, Reagan, Luke Miratrix, Aaron Russell Kaufman, and L Jason Anastasopoulos. 2018. "Matching with Text Data: An Experimental Evaluation of Methods for Matching Documents and of Measuring Match Quality." *arXiv Preprint arXiv:1801.00644.*

Nielsen, Richard A. 2013. "Rewarding Human Rights? Selective Aid Sanctions Against Repressive States." *International Studies Quarterly* 57 (4). Blackwell Publishing Ltd Oxford, UK: 791–803.

———. 2017. *Deadly Clerics: Blocked Ambition and the Paths to Jihad.* Cambridge University Press.

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.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. 2014. "GloVe: Global Vectors for Word Representation." In *Empirical Methods in Natural Language Processing (Emnlp)*, 1532–43. http://www.aclweb.org/anthology/D14-1162.

Quinn, Kevin M, Burt L Monroe, Michael Colaresi, Michael H Crespin, and Dragomir R Radev. 2010. "How to Analyze Political Attention with Minimal Assumptions and Costs." *American Journal of Political Science* 54 (1). Wiley Online Library: 209–28.

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). Sage Publications Sage UK: London, England: 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.

Roberts, Margaret E, Brandon M Stewart, and Richard A Nielsen. 2018. "Adjusting for Confounding with Text Matching."

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.

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, 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.

Slapin, Jonathan B, and Sven-Oliver Proksch. 2008. "A Scaling Model for Estimating Time-Series Party Positions from Texts." *American Journal of Political Science* 52 (3). Wiley Online Library: 705–22.

Spirling, Arthur. 2012. "US Treaty Making with American Indians: Institutional Change and Relative Power, 1784–1911." *American Journal of Political Science* 56 (1). Wiley Online Library: 84–97.

Stone, Philip J, Robert F Bales, J Zvi Namenwirth, and Daniel M Ogilvie. 1962. "The General Inquirer: A Computer System for Content Analysis and Retrieval Based on the Sentence as a Unit of Information." *Systems Research and Behavioral Science* 7 (4). Wiley Online Library: 484–98.

Tao, Ran, Daniel Strandow, Michael Findley, Jean-Claude Thill, and James Walsh. 2016. "A Hybrid Approach to Modeling Territorial Control in Violent Armed Conflicts." *Transactions in GIS* 20 (3). Wiley Online Library: 413–25.

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.

Weidmann, Nils B. 2016. "A Closer Look at Reporting Bias in Conflict Event Data." *American Journal of Political Science* 60 (1). Wiley Online Library: 206–18.

Wilkerson, John, and Andreu Casas. 2017. "Large-Scale Computerized Text Analysis in Political Science: Opportunities and Challenges." *Annual Review of Political Science* 20. Annual Reviews: 529–44.

Yang, Zichao, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J Smola, and Eduard H Hovy. 2016. "Hierarchical Attention Networks for Document Classification." In *HLT-Naacl*, 1480–9.