Advance Access publication January 22, 2013

Political Analysis (2013) 21:267–297
doi:10.1093/pan/mps028

Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts

Justin Grimmer
Department of Political Science, Stanford University, Encina Hall West 616 Serra Street, Stanford, CA 94305
e-mail: jgrimmer@stanford.edu (corresponding author)

Brandon M. Stewart
Department of Government and Institute for Quantitative Social Science, Harvard University, 1737 Cambridge Street, Cambridge, MA 02138
e-mail: bstewart@fas.harvard.edu

Edited by R. Michael Alvarez

Politics and political conflict often occur in the written and spoken word. Scholars have long recognized this, but the massive costs of analyzing even moderately sized collections of texts have hindered their use in political science research. Here lies the promise of automated text analysis: it substantially reduces the costs of analyzing large collections of text. We provide a guide to this exciting new area of research and show how, in many instances, the methods have already obtained part of their promise. But there are pitfalls to using automated methods—they are no substitute for careful thought and close reading and require extensive and problem-specific validation. We survey a wide range of new methods, provide guidance on how to validate the output of the models, and clarify misconceptions and errors in the literature. To conclude, we argue that for automated text methods to become a standard tool for political scientists, methodologists must contribute new methods and new methods of validation.

1 Introduction

Language is the medium for politics and political conflict. Candidates debate and state policy positions during a campaign. Once elected, representatives write and debate legislation. After laws are passed, bureaucrats solicit comments before they issue regulations. Nations regularly negotiate and then sign peace treaties, with language that signals the motivations and relative power of the countries involved. News reports document the day-to-day affairs of international relations that provide a detailed picture of conflict and cooperation. Individual candidates and political parties articulate their views through party platforms and manifestos. Terrorist groups even reveal their preferences and goals through recruiting materials, magazines, and public statements.

These examples, and many others throughout political science, show that to understand what politics is about we need to know what political actors are saying and writing. Recognizing that language is central to the study of politics is not new. To the contrary, scholars of politics have long recognized that much of politics is expressed in words. But scholars have struggled when using texts to make inferences about politics. The primary problem is volume: there are simply *too many* political texts. Rarely are scholars able to manually read all the texts in even moderately sized corpora. And hiring coders to manually read all documents is still very expensive. The result is that

Authors' note: For helpful comments and discussions, we thank participants in Stanford University's Text as Data class, Mike Alvarez, Dan Hopkins, Gary King, Kevin Quinn, Molly Roberts, Mike Tomz, Hanna Wallach, Yuri Zhurkov, and Frances Zlotnick. Replication data are available on the *Political Analysis* Dataverse at http://hdl.handle.net/1902.1/18517. Supplementary materials for this article are available on the *Political Analysis* Web site.

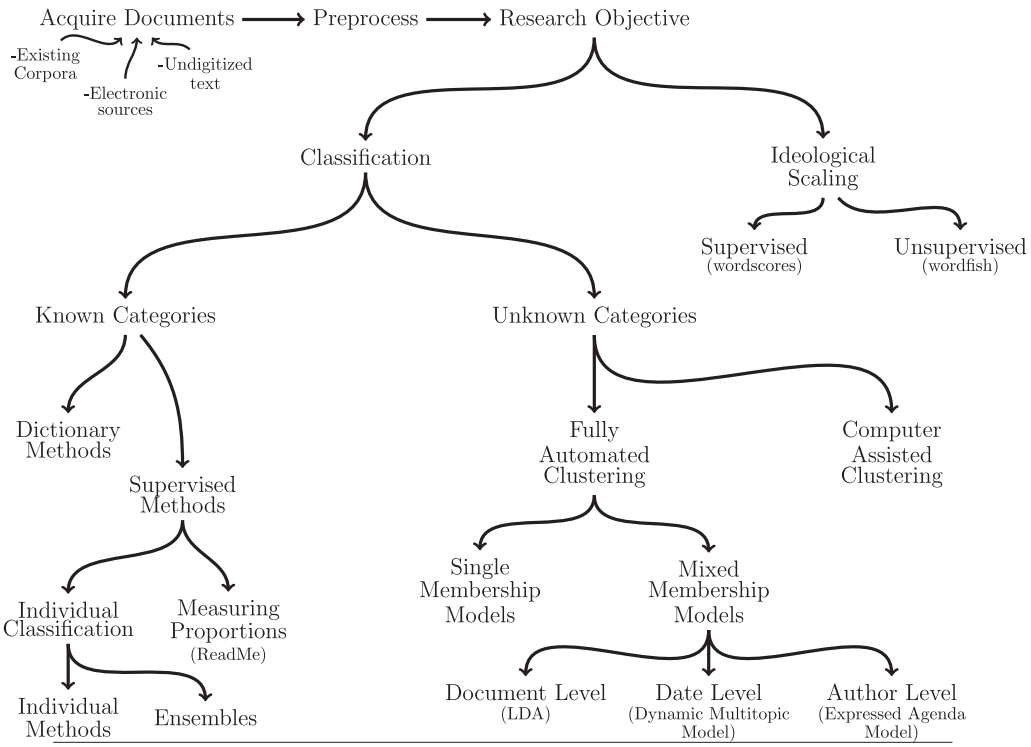

Fig. 1 An overview of text as data methods.

analyzing massive collections of text has been essentially impossible for all but the most well-funded projects.

We show how automated content methods can make possible the previously impossible in political science: the systematic analysis of large-scale text collections without massive funding support. Across all subfields of political science, scholars have developed or imported methods that facilitate substantively important inferences about politics from large text collections. We provide a guide to this exciting area of research, identify common misconceptions and errors, and offer guidelines on how to use text methods for social scientific research.

We emphasize that the complexity of language implies that automated content analysis methods will never replace careful and close reading of texts. Rather, the methods that we profile here are best thought of as *amplifying* and *augmenting* careful reading and thoughtful analysis. Further, automated content methods are *incorrect* models of language. This means that the performance of any one method on a new data set cannot be guaranteed, and therefore validation is essential when applying automated content methods. We describe best practice validations across diverse research objectives and models.

Before proceeding we provide a road map for our tour. Figure 1 provides a visual overview of automated content analysis methods and outlines the process of moving from collecting texts to applying statistical methods. This process begins at the top left of Fig. 1, where the texts are initially collected. The burst of interest in automated content methods is partly due to the proliferation of easy-to-obtain electronic texts. In Section 3, we describe document collections which political scientists have successfully used for automated content analysis and identify methods for efficiently collecting new texts.

With these texts, we overview methods that accomplish two broad tasks: classification and scaling. *Classification* organizes texts into a set of categories. Sometimes researchers know the categories beforehand. In this case, automated methods can minimize the amount of labor needed to classify documents. *Dictionary* methods, for example, use the frequency of key words to determine a document's class (Section 5.1). But applying dictionaries outside the domain for which they were developed can lead to serious errors. One way to improve upon dictionaries are

supervised methods (Section 5.2). These methods begin with human hand coding of documents into a predetermined set of categories. The human hand coding is then used to train, or *supervise* statistical models to classify the remaining press releases. But the performance of any one classifier can vary substantially across contexts—so validation of a classifier's accuracy is essential to establish the reliability of supervised learning methods.

Classification methods can also be used to discover new ways of organizing texts. *Fully Automated Clustering* (FAC) algorithms simultaneously estimate the categories and then classify documents into those categories (Section 6.1). Applying FAC algorithms to texts may have mixed results—it is difficult to know *a priori* if any one method will provide useful clusterings in a particular context. We describe two approaches to improve the output of FAC algorithms. The first set of models, *mixed membership models*, include problem-specific structure to assist in the estimation of categories (Section 6.1.2). The second model, *Computer Assisted Clustering (CAC)*, provides a method to explore thousands of potential clusterings (Section 6.2). Because there is no guarantee that unsupervised methods will return classes that are theoretically interesting, validation is essential. We describe several validation methods, including validations that combine supervised and unsupervised learning methods (Section 6.4).

Automated content methods can also estimate the location of actors in policy space, or produce a *scaling*. One method, *word scores*, relies on guidance from reference texts to situate other political actors in a space (Section 7.1). A second method, *word fish*, exploits an assumption about how ideology affects word usage (Section 7.2). We show how both methods rely heavily on the assumption that ideology dominates the language used in the text. When this assumption fits, when political actors are expressing their policy positions in texts, the models can perform well. When this assumption is violated, the model will place actors in a different and non-ideological space. We show that this space can be useful, but validations are necessary to establish its meaning.

No one article can hope to survey such a massive field. We have focused on methods that satisfy a diverse set of research objectives, focusing specifically on methods that analyze texts at the *document* level. This is most apparent with the absence of natural language processing methods to parse sentences and understand events (Jurafsky and Martin 2009). These methods are certainly useful in other fields, but have yet to gain widespread use in political science. The one notable exception is the work on events data collection (which has already been well-reviewed in Schrodt 2006). In the conclusion (and Supplementary Material), we direct readers to further sources to learn more about the methods introduced in this article and methods that we regretfully exclude. We also caution the reader that the space allocated to methods in this article is not necessarily in proportion to their use among political scientists. Rather, some methods—such as supervised learning methods—are relatively conceptually simple and require less validation. Other methods, such as unsupervised classification methods, require much more explanation and greater space for validation.

2 Four Principles of Automated Text Analysis

This section presents our four principles of automated content analysis methods (summarized in Table 1). We provide a brief introduction to each principle here. Throughout our exploration of the methods, we will revisit these principles and see that they offer a useful guide for using and evaluating quantitative methods for content analysis.

2.1 *Principle 1: All Quantitative Models of Language Are Wrong—But Some Are Useful*

The data generation process for any text (including this one) is a mystery—even to linguists. If any one sentence has complicated dependency structure, its meaning could change drastically with

Table 1 Four principles of quantitative text analysis

(1) All quantitative models of language are wrong—but some are useful.
(2) Quantitative methods for text amplify resources and augment humans.
(3) There is no globally best method for automated text analysis.
(4) Validate, Validate, Validate.

the inclusion of new words, and the sentence context could drastically change its meaning. Consider, for example, the classic joke "Time flies like an arrow. Fruit flies like a banana." This joke relies on the extreme complexity of parsing the sentence, using the change in our understanding of "flies like" as the sentence progresses—from metaphorical to literal.

The complexity of language implies that all methods *necessarily* fail to provide an accurate account of the data-generating process used to produce texts. Automated content analysis methods use insightful, but wrong, models of political text to help researchers make inferences from their data. The goal of building text models is therefore different than model building to make causal inferences—the primary goal of most political science models. The usual advice for causal inference model building is that it is essential to include all the relevant features of the data-generating process—either in covariates or in the model structure (e.g., Greene 2007). But the same advice does not hold for quantitative text analysis methods. Including more realistic features into quantitative models does not necessarily translate into an improved method, and reducing the assumptions used may not imply more productive analyses. Rather, subtleties of applying the methods to any one data set mean that models that are less sophisticated in the use of language may provide more useful analysis of texts.

That all automated methods are based on incorrect models of language also implies that the models should be evaluated based on their ability to perform some useful social scientific task. As we explain below, emphasis in evaluations should be placed on helping researchers to assign documents into predetermined categories, discover new and useful categorization schemes for texts, or in measuring theoretically relevant quantities from large collections of text. Alternative model evaluations that rely on model fit or predicting the content of new texts can select substantively inferior models (see Chang et al. 2009).

2.2 *Principle 2: Quantitative Methods Augment Humans, Not Replace Them*

Automated content analysis methods have demonstrated performance across a variety of substantive problems. These methods will not, however, eliminate the need for careful thought by researchers nor remove the necessity of reading texts. Indeed a deep understanding of the texts is one of the key advantages of the social scientist in applying automated methods. We will see below that researchers still guide the process, make modeling decisions, and interpret the output of the models. All these require the close reading of texts and thoughtful analysis by the researcher.

Rather than replace humans, computers *amplify* human abilities. The most productive line of inquiry, therefore, is not in identifying how automated methods can obviate the need for researchers to read their text. Rather, the most productive line of inquiry is to identify the best way to use both humans and automated methods for analyzing texts.

2.3 *Principle 3: There Is No Globally Best Method for Automated Text Analysis*

Different data sets and different research questions often lead to different quantities of interest. This is particularly true with text models. Sometimes, the goal is to identify the words that distinguish the language of groups (Laver, Benoit, and Garry 2003; Monroe, Colaresi, and Quinn 2008). In other research projects, the quantity of interest is the proportion of documents that fall within a predetermined set of categories (Hopkins and King 2010). And in yet other research projects, scholars may want to use quantitative methods to discover a new way to organize texts (e.g., Grimmer and King 2011) or to locate actors in a spatial model (Laver, Benoit, and Garry 2003; Monroe and Maeda 2004; Slapin and Proksch 2008).

Each of the research questions imply different *models*, or families of models, to be used for analysis, and different methods of validation. Therefore, much of the debate across approaches for automated text analysis is misguided. We show below that much of the across-approach debate can be resolved simply by acknowledging that there are different research questions and designs that imply different types of models will be useful. Rather than debating across approaches to text analysis, we think one of the most important lines of inquiry is identifying effective ways to combine previously distinct methods for quantitative text analysis.

There is also substantial variation within families of models. Perhaps unsurprisingly, the same model will perform well on some data sets, but will perform poorly when applied to other data. Therefore, establishing one method to always use for one task is almost guaranteed to be impossible. Instead scholars will need to carefully think and apply different methods to generate useful and acceptable estimates for their problem.

2.4 *Principle 4: Validate, Validate, Validate*

Automated text analysis methods can substantially reduce the costs and time of analyzing massive collections of political texts. When applied to any one problem, however, the output of the models may be misleading or simply wrong. Therefore, it is incumbent upon the researcher to *validate* their use of automated text analysis. This validation can take several forms, many of which we describe below. When categories are known in a calculation problem, scholars must demonstrate that the supervised methods are able to reliably replicate human coding. Validation for unsupervised methods is less direct. To validate the output of an unsupervised method, scholars must combine experimental, substantive, and statistical evidence to demonstrate that the measures are as conceptually valid as measures from an equivalent supervised model. Similar to unsupervised methods, validating ideological scalings requires numerous and substance-based evaluations (Budge and Pennings 2007; Slapin and Proksch 2008).

What should be avoided, then, is the blind use of any method without a validation step. This is why we discourage the use of many commercial tools for quantitative text analysis. These programs simply provide the researcher with output. It is often difficult, and sometimes impossible, to validate the output. But more damning for these methods is that once a problem is identified, it is exceedingly difficult to change how the analysis is conducted. Certainly, these methods can be validated and provide conceptually valid and useful output. But without a broader set of tools available, it is difficult to know if the methods included in the commercially available software are optimal for the particular problem at hand.

Before applying these principles, texts are needed. The next section describes common sources for data and characteristics of data that make automated methods more likely to succeed.

3 Acquiring Text

Political scientists have applied automated content analysis across a diverse set of texts. This includes archives of media data (Young and Soroka 2011); floor speeches in legislatures from across the world (Quinn et al. 2010); presidential, legislator, and party statements (Grimmer 2010); proposed legislation and bills (Adler and Wilkerson 2011); committee hearings (Jones, Wilkerson, and Baumgartner 2009); treaties (Spirling 2012); political science papers; and many other political texts.

This explosion is partly due to the rapid move to store and distribute documents in electronic text databases. Automated methods require texts stored in a plain text format—UTF-8 for Latin characters, for example. The easiest way to acquire text in this form is from online databases of articles. Lexis Nexis and ProQuest, for example, facilitate batch downloads of files, and J STOR provides already processed texts of academic articles. More sources are being added all the time; for example, the U.S. House of Representatives recently launched a new Web site dedicated to the distribution of all current House Resolutions under discussion in Xstensible Markup Language (XML) format, an easily accessible and widely used method for storing structured text.

Slightly more difficult to access are text data stored on Web sites, but automated scraping methods make acquiring these data easier (Jackman 2006). And even when scraping data is not an option—perhaps due to Web site restrictions—online platforms that distribute tasks across workers, such as Amazon's Mechanical Turk, can make acquiring data efficient (Berinsky, Huber, and Lenz 2012). The most difficult to acquire are texts found in archives or yet-to-be-scanned books. But preparing these texts for analysis can be straightforward—using a high-quality scanner and Optical Character Recognition software, it is possible to convert archival

materials into computer readable texts (see e.g., the data collection process in Eggers and Hainmueller 2009).

Texts abound in politics and automated methods are potentially useful across most political texts. But some texts better fit the assumptions of the automated content methods described here. These methods tend to work better when the text is focused—either on the expression of one idea for classification or the expression of policy positions for scaling methods. In some instances, discarding text not related to the primary quantity of interest can actually improve the performance of automated clustering methods. Automated methods also rely on having a sufficient number of words to use reliably. This makes lengthier texts—newspapers or party platforms—often much easier to analyze than shorter statements, such as open-ended survey responses. For shorter texts, accompanying information (or an extremely large volume of texts) is often necessary for classification or scaling methods to perform reliably.

With texts in hand, the next section shows how to move words on the page to numbers analyzed statistically.

4 Reducing Complexity: From Words to Numbers

Language is complex. But not all of language's complexity is necessary to effectively analyze texts. In this section, we present a recipe for transforming text into quantitative data. Each of the steps are designed to retain information that will be used by the automated methods, while discarding information that will likely be unhelpful, ancillary, or too complex for use in a statistical model. The recipe is easy to apply, with several pieces of freely available software able to apply the sequence of steps we describe here. We present a recipe for representing text quantitatively, but emphasize that any one of the steps that we present here can—and oftentimes should—be modified. More important than following any individual recipe is to think carefully about the particular problem at hand, test different approaches, and validate the results.

Throughout, we will refer to the unit of analysis as a *text* or *document*, but it could apply to any unit of text: a tweet, Facebook status, spoken utterance, press briefing, sentence, or paragraph. We refer to the population of texts to be analyzed as the *corpus* and a collection of these as *corpora*. Some methods will work better on different types of tasks or documents of different lengths, but most methods begin in the same way with a series of preprocessing steps to reduce the awe inspiring diversity of language to a manageable set of features.

The most consequential, and shocking, step we will take is to discard the order in which words occur in documents (Jurafsky and Martin 2009). We will assume documents are a *bag of words*, where order does not inform our analyses. While it is easy to construct sample sentences where word order fundamentally changes the nature of the sentence, empirically these sentences are rare. A simple list of words, which we call *unigrams*, is often sufficient to convey the general meaning of a text. If this assumption is unpalatable, we can retain some word order by including *bigrams* (word pairs) or *trigrams* (word triples) into our analysis (Jurafsky and Martin 2009). This allows us to distinguish, for example, the "White House" from the color and the domicile. In practice, for common tasks like measuring sentiment, topic modeling, or search, *n-grams* do little to enhance performance (Manning, Raghavan, and Schütze 2008; Hopkins and King 2010).

After discarding word order, we simplify the vocabulary with *stemming*. Stemming removes the ends of words to reduce the total number of unique words in the data set, or reduce the *dimensionality* of text. Stemming reduces the complexity by mapping words that refer to the same basic concept to a single root. For example, `family`, `families`, `families'`, and `familial` all become *famili*. Stemming is actually an approximation to a linguistic concept called *lemmatization*, which seeks to reduce words to their base forms (Manning, Raghavan, and Schütze 2008; Jurafsky and Martin 2009). The critical difference is that a lemmatizer uses context and dictionaries to help discover (for example) that good is the base form of `better` and `best`. The stemmer is a much cruder algorithm, but considerably faster. The performance does not seem to matter for most applications, so the majority of applied research uses stemming. There are numerous stemming algorithms available that vary in the extent and frequency of word truncation.

But the Porter stemming algorithm (Porter 1980) is commonly employed because of its moderate approach to word simplification.

In addition to discarding word order, we also typically discard punctuation, capitalization, very common words (often we remove "stop" words, or function words that do not convey meaning but primarily serve grammatical functions), and very uncommon words (words that appear only once or twice in the corpus and thus are unlikely to be discriminating). We typically remove words which appear in less than 1% and more than 99% of documents in the corpus, although these choices need to be made contingent both on the diversity of the vocabulary, average length of the document, and the size of the corpus (Quinn et al. 2010; Hopkins and King 2010).

The result of the preprocessing steps is that each document i ($i = 1, \ldots, N$) is represented as a vector that counts the number of times each of the M unique words occur, $W_i = (W_{i1}, W_{i2}, \ldots, W_{iM})$. Each W_{im} counts the number of times the *m-th* word occurs in the *i-th* document. The collection of count vectors into a matrix is often called the *document-term matrix*. For a moderate volume of documents without a particularly specialized vocabulary, this matrix will have between three thousand and five thousand *features* or terms and will contain mostly zeroes (a condition we call *sparsity*).

These steps seem to result in a shocking reduction of information, leaving many to conclude that there will be too little information to extract anything meaningful from the texts. But consistently across applications, scholars have shown that this simple representation of text is sufficient to infer substantively interesting properties of texts (Hopkins and King 2010).

4.1 *Alternative Methods for Representing Text*

The recipe described above is one way to represent text as data. This can, and should, be varied as needed for specific applications. For example, in one of the first examples of quantitative text analysis, Mosteller and Wallace (1963) sought to infer the authorship of the unattributed Federalist Papers. Since they were interested specifically in the style of the documents and not their content, they used only the counts of function words or *stopwords*. Thus, their entire analysis relied on information that we customarily discard. Some other common strategies include: (1) using an indicator that a word occurs in a document, rather than a count (Pang, Lee, and Vaithyanathan 2002; Hopkins and King 2010); (2) including some commonly used stopwords such as gendered pronouns (Monroe, Colaresi, and Quinn 2008); (3) a subset of features (either by automated feature selection or a lower dimensional projection) (Hofmann 1999); and (4) weighting words by their rarity in the document set (often called *tf-idf* or term frequency by inverse document frequency weighting) (Manning, Raghavan, and Schütze 2008).

Although these are fundamentally similar ways of describing the same basic feature set (the list of unordered unigrams), sometimes the problem calls for a completely different approach. Spirling (2012), for example, analyzes treaties between the Native Americans and the U.S. government. In this case, discarding word order would mask information on land negotiations. In order to preserve word order information, Spirling (2012) uses *sub-string kernels*. The sub-string portion means that each feature is a small sequence of letters (e.g., five) that can span multiple words. Since using this feature space would be unimaginably large, Spirling (2012) uses technology from Lodhi et al. (2002) to calculate only the distance between the documents in this feature space using only sub-strings that occur in both documents. Thus, scaling performed on the resulting matrix of distances is able to account for word order.

5 Classifying Documents into Known Categories

Assigning texts to categories is the most common use of content analysis methods in political science. For example, researchers may ask if campaigns issue positive or negative advertisements (Ansolabehere and Iyengar 1995), if legislation is about the environment or some other issue area (Adler and Wilkerson 2011), if international statements are belligerent or peaceful (Schrodt 2000), or if local news coverage is positive or negative (Eshbaugh-Soha 2010). In each instance, the goal is

to infer either the category of each document, the overall distribution of documents across categories, or both.

Human-based methods for making these inferences are both time and resource intensive. Even after coding rules are developed and coders are trained, manual methods require that coders read each individual text. Automated methods can mitigate the cost of assigning documents to categories, by limiting the amount of classification humans perform. We characterize two broad groups of methods for reducing the costs of classification. *Dictionary* methods use the relative frequency of *key words* to measure the presence of each category in texts. Supervised learning methods replicate the familiar manual coding task, but with a machine. First, human coders are used to classify a subset of documents into a predetermined categorization scheme. Then, this *training set* is used to train an automated method, which then classifies the remaining documents.

5.1 *Dictionary Methods*

We begin with dictionary methods, perhaps the most intuitive and easy to apply automated method (Stone et al. 1966). Dictionaries use the rate at which key words appear in a text to classify documents into categories or to measure the extent to which documents belong to particular categories. For example, suppose the goal is to measure the *tone* in newspaper articles (e.g., Eshbaugh-Soha 2010): whether articles convey information positively or negatively. Dictionary methods use a list of words with attached tone scores and the relative rate at which words occur to measure a document's tone. A *dictionary* to measure tone is a list of words that are either dichotomously classified as positive or negative or contain more continuous measures of their content. Formally, each word m ($m = 1, \ldots, M$) will have associated score s_m. For the simplest measures, $s_m = -1$ if the word is associated with a negative tone and $s_m = 1$ if associated with a positive tone. If $N_i = \sum_{m=1}^{M} W_{im}$ words are used in document i, then dictionary methods can measure the tone for any document t_i as,

$$t_i = \sum_{m=1}^{M} \frac{s_m W_{im}}{N_i}.$$

Scholars often use t_i as an approximately continuous measure of document tone, but it also can be used to classify documents into tone categories if a decision rule is assumed along with the dictionary method. Perhaps the simplest coding rule would assign all documents with $t_i > 0$ to a positive tone category and $t_i < 0$ to a negative tone.

Tone is just one type of analysis a dictionary method can perform. The general idea of dictionaries make them relatively easy and cheap to apply across a variety of problems: identify words that separate categories and measure how often those words occur in texts (for some recent examples that use dictionaries to measure a variety of concepts, see Kellstedt 2000; Laver and Garry 2000; Burden and Sanberg 2003; Young and Soroka 2011). Finding the separating words is also relatively easy. There are a variety of widely used off-the-shelf dictionaries that provide key words for a variety of categories (e.g., Bradley and Lang 1999; Hart 2000; Pennebaker, Francis, and Booth 2001; Turney and Littman 2003). And if scholars have documents already coded into categories, dictionaries can be produced using existing methods. Monroe, Colaresi, and Quinn (2008) describe a variety of methods that measure how well words separate already identified categories of interest (see also Taddy 2010 and Diermeier et al. 2011). Any one of these methods could be used to produce dictionary-like scores of words, which could then be applied in other contexts to classify documents.

For dictionary methods to work well, the scores attached to words must closely align with how the words are used in a particular context. If a dictionary is developed for a specific application, then this assumption should be easy to justify. But when dictionaries are created in one substantive area and then applied to another, serious errors can occur. Perhaps the clearest example of this is shown in Loughran and McDonald (2011). Loughran and McDonald (2011) critique the increasingly common use of off-the-shelf dictionaries to measure the tone of statutorily required corporate earning reports in the accounting literature. They point out that many words that have a negative

connotation in other contexts, like `tax`, `cost`, `crude` (oil), or `cancer`, may have a positive connotation in earning reports. For example, a health-care company may mention `cancer` often and oil companies are likely to discuss `crude` extensively. And words that are not identified as negative in off-the-shelf dictionaries may have quite negative connotation in earning reports (e.g., `unanticipated`).

Dictionaries, therefore, should be used with substantial caution, or at least coupled with explicit validation. When applying dictionaries, scholars should directly establish that word lists created in other contexts are applicable to a particular domain, or create a problem-specific dictionary. In either instance, scholars must validate their results. But measures from dictionaries are rarely validated. Rather, standard practice in using dictionaries is to assume the measures created from a dictionary are correct and then apply them to the problem. This is due, in part, to the exceptional difficulties in validating dictionaries. Dictionaries are commonly used to establish granular scales of a particular kind of sentiment, such as tone. Although this is useful for applications, humans are unable to produce the same granular measures reliably (Krosnick 1999). The result is that it is essentially impossible to derive gold-standard evaluations of dictionaries based on human coding of documents.

The consequence of domain specificity and lack of validation is that most analyses based on dictionaries are built on shaky foundations. Yes, dictionaries are able to produce measures that are claimed to be about tone or emotion, but the actual properties of these measures—and how they relate to the concepts they are attempting to measure—are essentially a mystery. Therefore, for scholars to effectively use dictionary methods in their future work, advances in the validation of dictionary methods must be made. We suggest two possible ways to improve validation of dictionary methods. First, the classification problem could be simplified. If scholars use dictionaries to code documents into binary categories (e.g., positive or negative tone), then validation based on human gold standards and the methods we describe in Section 5.2.3 is straightforward. Second, scholars could treat measures from dictionaries similar to how validations from unsupervised methods are conducted (see Section 6.4). This would force scholars to establish that their measures of underlying concepts satisfy several different standards of validity.

5.2 *Supervised Learning Methods*

Dictionary methods require scholars to identify words that separate classes beforehand. This can lead to inefficiencies in applying dictionaries to real data problems—particularly if dictionaries are applied outside of the domain for which they were originally developed. Supervised learning methods provide a useful alternative method for assigning documents to predetermined categories. The idea of supervised learning is simple: human coders categorize a set of documents by hand. The algorithm then "learns" how to sort the documents into categories using the training set and words: the algorithm uses characteristics of the documents to place the documents into the categories.

This approach to classification has two major advantages over dictionary methods. First, it is necessarily domain specific and therefore avoids the problems of applying dictionaries outside of their intended area of use. Applying supervised learning methods requires scholars to develop coding rules for the particular quantities of interest. The model is then trained using a sample of documents from the corpus that is to be classified. This also forces scholars to develop coherent definitions of concepts for particular applications, which leads to clarity in what researchers are measuring and studying. Second, supervised learning methods are much easier to validate, with clear statistics that summarize model performance.

Supervised methods for text classification is a massive—and rapidly expanding—area of research (see, e.g., the excellent software provided in Jurka et al. 2012). But all supervised learning methods share three basic steps: (1) construct a training set; (2) apply the supervised learning method—learning the relationship between features and categories in the training set, then using this to infer the labels in the test set; and (3) validate the model output and classify the remaining documents.

We outline each of the steps here—how to construct a training set, training and applying statistical models, and how to validate and assess the performance of the method.

5.2.1 Constructing a training set

The most important step in applying a supervised learning algorithm is constructing a reliable training set, because no statistical model can repair a poorly constructed training set, and well-coded documents can hide faults in simple models. We divide the construction of the training set into two components: (1) creating and executing a coding scheme and (2) sampling documents.

Creating a Coding Scheme: Ambiguities in language, limited attention of coders, and nuanced concepts make the reliable classification of documents difficult—even for expert coders. To address this difficulty, best practice is to iteratively develop coding schemes. Initially, a concise codebook is written to guide coders, who then apply the codebook to an initial set of documents. When using the codebook, particularly at first, coders are likely to identify ambiguities in the coding scheme or overlooked categories. This subsequently leads to a revision of the codebook, which then needs to be applied to a new set of documents to ensure that the ambiguities have been sufficiently addressed. Only after coders apply the coding scheme to documents without noticing ambiguities is a "final" scheme ready to be applied to the data set.

Creating coding schemes is a rich literature, with contributions across social science fields. For more on the schemes—including how to assess coder agreement and practical guides to scheme creation—see Krippendorff (2004); Neuendorf (2002); Weber (1990) and the documentation available in the `ReadMe` software (Hopkins et al. 2010).

Selecting Documents: Ideally, training sets should be representative of the corpus. Supervised learning methods use the relationship between the features in the training set to classify the remaining documents in the test set. In fact, almost all classification methods implicitly assume that the training set is a random sample from the population of documents to be coded (Hand 2006). Given this assumption, it is not surprising that best performance comes from random sampling to obtain a representative sample of documents—either through simple random sampling or from a more complicated stratified design. This may seem obvious, but presents particular difficulty when all the data are not available at the time of coding: either because it will be produced in the future or because it has yet to be digitized.

Training sets also need enough documents to apply supervised methods accurately. Hopkins and King (2010) offer five hundred as a rule of thumb with one hundred documents probably being enough. This is generally useful guidance, though it can be dangerous to apply a strict rule when selecting the number of documents. The number necessary will depend upon the specific application of interest. For example, as the number of categories in a coding scheme increases, the number of documents needed in the training set also increases. Supervised methods need enough information to learn the relationship between words and documents in *each* category of a coding scheme.[1]

5.2.2 Applying a supervised learning model

After hand classification is complete, the hand-labeled documents are used to train the supervised learning methods to learn about the test set—either classifying the individual documents into categories or measuring the proportion of documents in each category. The methods to do this classification are diverse, though they share a common structure that usefully unifies the methods (Hastie, Tibshirani, and Friedman 2001).

To see this common structure, suppose there are N_{train} documents ($i = 1, \ldots, N_{\text{train}}$) in our training set and each has been coded into one-of-K categories ($k = 1, \ldots, K$). Each document i's category is represented by $Y_i \in \{C_1, C_2 \ldots, C_K\}$ and the entire training set is represented as $\boldsymbol{Y}_{\text{train}} = (Y_1, \ldots, Y_{N_{\text{train}}})$. Recall from Section 4 that each document i's features are contained in an M length vector $\boldsymbol{W}_i$, which we collect in the $N_{\text{train}} \times M$ matrix $\boldsymbol{W}_{\text{train}}$. Each supervised learning

[1] Determining the number of documents necessary to code documents can be challenging, but this need not be a blind search: the validation schemes that we use in Section 5.2.3 can be applied to determine the return on coding more documents.

algorithm assumes that there is some (unobserved) function that describes the relationship between the words and the labels,

$$\boldsymbol{Y}_{\text{train}} = f(\boldsymbol{W}_{\text{train}}).$$

Each algorithm attempts to learn this relationship—estimating the function f with $\hat{f}$. $\hat{f}$ is then used to infer properties of the test set, $\hat{Y}_{\text{test}}$—either each document's category or the overall distribution of categories—using the test set's words $\boldsymbol{W}_{\text{test}}$,

$$\widehat{\boldsymbol{Y}}_{\text{test}} = \hat{f}(\boldsymbol{W}_{\text{test}}).$$

We now overview three methods for inferring the relationship between words and categories.

Individual Methods: To provide a sense of how individual classification algorithms work, we present in detail a canonical example—the Naïve Bayes classifier (Maron and Kuhns 1960). The model has a simple, but powerful, approach to learning the relationship between words and categories. The training set is used to learn about the distribution of words for documents from category k. This distribution is then used to classify each of the documents in the test set. To perform this classification, Naïve Bayes begins with Bayes's rule. The goal is to infer the probability that document i belongs to category k given word profile $\boldsymbol{W}_i$. Applying Bayes's rule,

$$p(C_k|\boldsymbol{W}_i) \propto p(C_k)p(\boldsymbol{W}_i|C_k) \tag{1}$$

where we drop $p(\boldsymbol{W}_i)$ from the denominator since we know that it is a constant across the different categories. Equation (1) shows that to estimate $p(C_k|\boldsymbol{W}_i)$, we need estimates of $p(C_k)$ and $p(\boldsymbol{W}_i|C_k)$. If the documents in the training set are representative of the corpus, then the maximum likelihood estimate of $p(C_k)$ is straightforward:

$$\hat{p}(C_k) = \frac{\text{Number Train Docs in Category } k}{N_{\text{train}}}$$

or the proportion of documents from the training set in each category. Estimating $p(\boldsymbol{W}_i|C_k)$ is more challenging, because of the large number of words used to represent each text. For even moderately sized texts this implies that any one vector of word counts $\boldsymbol{W}_i$ will only appear in a corpus once—implying that $\hat{p}(\boldsymbol{W}_i|C_k) = 0$ for all i observations in the test set and all k categories. Therefore, without additional assumptions, the model is useless for classification. To make a useful estimation of $p(\boldsymbol{W}_i|C_k)$ possible, we introduce the "naïve" assumption in the model: the model assumes that, given a document's category, the words are generated independently,

$$p(\boldsymbol{W}_i|C_k) = \prod_{i=1}^{M} p(W_{im}|C_k).$$

Of course, this assumption must be wrong: the use of words is highly correlated any data set. But even though the assumption is wrong, the model is still able to capture information in texts useful for classification. Using this assumption, estimation of $p(W_{im}|C_k)$ is straightforward,

$$\hat{p}(W_{im} = j|C_k) = \frac{\text{Number Train Docs in Category } k \text{ and With Word } m \text{ Used } j \text{ times}}{\text{Number Train Docs in Category } k}.$$

This simplified model still presents challenges, because some word-specific counts never occur in the data set. The common solution is to add a small amount to each probability, which is usually justified using a Bayesian Dirichlet-Multinomial model (Jurafsky and Martin 2009). Using the estimated components of the right-hand side of equation (1), Naïve Bayes then assigns each document in the test set to the document with the highest probability. Therefore, the estimated classifier for Naïve Bayes, $\hat{f}$ is,

$$\hat{f}(\boldsymbol{W}_i) = \arg\max_k \left[\hat{p}(C_k) \prod_{i=1}^{M} \hat{p}(W_{im}|C_k) \right]$$

The Naïve Bayes's classifier conforms neatly to our first principle: although the model is clearly wrong—of course features are not conditionally independent—it has proven to be a useful classifier for a diverse set of tasks (Hastie, Tibshirani, and Friedman 2001). But the Naïve Bayes's classifier is just one example of a very large and diverse literature, including Random Forests (Breiman 2001), Support Vector Machines (Venables and Ripley 2002), and neural networks (Bishop 1995).

Ensembles: On their own, individual methods for classification can provide accurate and reliable classifications. But it is also straightforward to combine classification models to produce a classifier which is superior to any of the individual classifiers. Heuristically, as long as the classifiers are accurate and diverse, combining the classifiers will improve accuracy (Jurafsky and Martin 2009). But ensembles are also useful for other reasons, including: increased out-of-sample stability and the ability to capture complex functional forms with relatively simple classifiers (Dietterich 2000; Hillard, Purpura, and Wilkerson 2008). Schemes for developing ensembles are diverse. For example, *Super-learning* uses cross-validation to assign weights to methods proportional to their out-of-sample accuracy (van der Laan, Polley, and Hubbard 2007). Additional methods include bagging—repeatedly drawing a sample with replacement of the training data and classifying the out of sample cases—and boosting—sequential training of classifiers increasing weight on misclassified cases (Hastie, Tibshirani, and Friedman 2001).

Measuring Proportions: A different way to improve the results is to change the quantity of interest. For many social science applications, only the proportion of documents in a category is needed—not the categories of each individual document. Shifting focus to estimating proportions, $P(\mathbf{C})$, can lead to substantial improvements in accuracy—even if the documents are not randomly sampled from the corpus (Hopkins and King 2010). The result is that *ReadMe* can provide reliable estimates of proportions across many domains and applications. To introduce ReadMe, we first modify the recipe described in Section 4, including an indicator of whether a word occurred in a document, rather than counts of the words. Using this representation, define a test-set-specific probability distribution over all possible documents, $p(\boldsymbol{W}_{\text{test}})$. Without further assumptions, the data-generating process for the test set can be written as,

$$p(\boldsymbol{W}_{\text{test}}) = p(\boldsymbol{W}_{\text{test}}|\boldsymbol{C}_{\text{test}})\, p(\boldsymbol{C}_{\text{test}}) \tag{2}$$

where $p(\boldsymbol{W}_{\text{test}}|\boldsymbol{C}_{\text{test}})$ is the distribution of documents in the test set conditional on categories and $p(C_{\text{test}})$ is the proportion of documents in each class in the test set—the quantity of interest. The most important insight is that solving for $p(\boldsymbol{C}_{\text{test}})$ is simple if $p(\boldsymbol{W}_{\text{test}})$ and $p(\boldsymbol{W}_{\text{test}}|\boldsymbol{C}_{\text{test}})$ are known. Of course, learning either quantities is the most challenging components of supervised learning: both quantities are high dimensional. One solution—used in Naïve Bayes—is to assume the words are generated independently. Hopkins and King (2010) avoid this assumption by employing matrix smoothing: this preserves higher order correlations between words, while also ensuring that the estimated probabilities are useful for learning from the training set. With this approach, $\hat{p}(\boldsymbol{W}_{\text{test}})$ can be estimated without coding any documents—it is inferred directly from the test set. Inferring $\hat{p}(\boldsymbol{W}_{\text{test}}|\boldsymbol{C}_{\text{test}})$ requires labeled documents—which are unavailable for the test set. But if we assume that the conditional distributions are identical in the training and test sets, then we can substitute $\hat{p}(\boldsymbol{W}_{\text{test}}|\boldsymbol{C}_{\text{test}})$ with $\hat{p}(\boldsymbol{W}_{\text{train}}|\boldsymbol{C}_{\text{train}})$. Heuristically, $\hat{f}$, used to estimate the proportion of documents in each category, is given by

$$\hat{f}(\boldsymbol{W}_{\text{test}}) = \left(\hat{p}(\boldsymbol{W}_{\text{train}}|\boldsymbol{C}_{\text{train}})'\hat{p}(\boldsymbol{W}_{\text{train}}|\boldsymbol{C}_{\text{train}})\right)^{-1}\hat{p}(\boldsymbol{W}_{\text{train}}|\boldsymbol{C}_{\text{train}})'\hat{p}(\boldsymbol{W}_{\text{test}}).$$

The move to proportions in Hopkins and King (2010) pays substantial dividends for inference—weakening assumptions used in other methods and reducing the potential bias systemic in other methods. But the `ReadMe` algorithm is not an option for all users, particularly those who have few documents or need to stratify the results by some external covariate. If there are a large number of documents (in excess of 100 at least) for each value of the covariate, the algorithm can simply be rerun on each strata, but otherwise it is necessary to return to individual classification.

Table 2 Confusion matrix: comparing human and supervised coding

		Training data		
		Restrained	*Activist*	*Neutral*
Machine	Restrained	111	31	28
	Activist	10	17	0
	Neutral	26	9	68

5.2.3 Validation

Supervised methods are designed to automate the hand coding of documents into categories or measuring the proportion of documents in categories. If a method is performing well, it will directly replicate the hand coding. If it performs poorly, it will fail to replicate the coding—instead introducing serious errors. This clear objective implies a clear standard for evaluation: comparing the output of machine coding to the output of hand coding. The ideal validation procedure would divide the data into three subsets. Initial model fitting would be performed on the training set. Once a final model is chosen, a second set of hand-coded documents—the validation set—would be used to assess the performance of the model. The final model would then be applied to the test to complete the classification.

This approach to validation is difficult to apply in most settings. But *cross-validation* can be used to replicate this ideal procedure (Efron and Gong 1983; Hastie, Tibshirani, and Friedman 2001). In V-fold cross-validation, the training set is randomly partitioned into V ($v = 1 \ldots, V$) groups. The model's performance is assessed on each of the groups, ensuring all predictions are made on data out of sample. For each group v, the model is trained on the $V - 1$ other groups, then applied to the *V-th* group to assess performance. Cross-validation is extremely powerful—it avoids overfitting by focusing on out-of-sample prediction and selects the best model for the underlying data from a set of candidate models (this is known as the Oracle property) (van der Vaart, Dudoit, and van der Laan 2006).[2]

5.2.4 Applying supervised learning methods: Russian military discourse

In order to demonstrate the use and validation of supervised learning methods, we adapt an example from Stewart and Zhukov (2009). To test a broader argument, Stewart and Zhukov (2009) compare the stances on foreign policy activism that civilian and military elites articulate in their public statements. They collect a corpus of 7920 Russian language public statements by political and military elites made between 1998 and 2008. Then following close reading of many of the documents, they develop a codebook to describe coding rules so that human coders could classify statements as having a restrained, activist, or neutral position on the Russian use of force.

Stewart and Zhukov (2009) then randomly sample and code three hundred documents. This seemingly low number is due to an additional constraint: finding and paying Russian-speaking coders substantially raised the costs of coding additional documents. With the set of human codings from Stewart and Zhukov (2009), we fit a *Random Forest* model to first learn the relationship between words and classes and then apply this relationship to classify the remaining documents.

To assess how well the Random Forest algorithm was able to replicate human coders, we performed a ten-fold cross-validation using the training data. This facilitates a direct comparison of machine and human classifications.

To summarize the model performance, Table 2 presents a *confusion* matrix. The rows of Table 2 are the out-of-sample codes from the Random Forest algorithm, the columns are the human-produced codes, and each cell entry counts the number of documents that received the

[2]These properties only apply when all steps (including selection of relevant predictors) are handled within the training set of the cross-validation and not within the full-labeled data. See Section 7.10.2 of Hastie, Tibshirani, and Friedman (2001) for more information on the proper application of cross-validation.

Table 3 Document classifications by Elite type (proportion in parentheses)

		Military	*Political*
Training set	Restrained	27 (0.36)	119 (0.53)
	Activist	25 (0.34)	32 (0.14)
	Neutral	22 (0.30)	74 (0.33)
Test set	Restrained	870 (0.41)	3550 (0.62)
	Activist	500 (0.24)	260 (0.04)
	Neutral	749 (0.35)	1960 (0.34)

corresponding Random Forest and human codes. So, for example, the top-left cell counts 111 statements that the supervised method and the human agreed were both restrained. The goal of the supervised learning method is to place as many documents as possible in the on-diagonal cells—or replicate the human coding.

We use three statistics to summarize the confusion matrix. The first, and most widely used is *accuracy*—the proportion of correctly classified documents. Computing we find an overall accuracy of,

$$\text{Accuracy} = \frac{\text{No. Doc. On Diagonal}}{\text{Total No. Doc.}} = 0.65.$$

This is not an extremely accurate classifier, though it is comparable to accuracy levels for complex categories found in other studies (Hopkins and King 2010). It is also common to measure *precision* for a category—given that the machine guesses category k, what is the probability that the machine made the right guess. Operationally, this is estimated as number of documents correctly classified into category k, divided by the total number of documents the machine classifies as category k. For restrained foreign policy positions, the precision is 0.65. The final statistic is *recall* for category k—given that a human coder labels a document as belonging to category k, what is the chance the machine identifies the document. This is calculated by taking the number of correctly classified category k documents divided by the number of human coded documents in category k. For the restrained category, the recall is 0.75. The differences between the precision and recall exemplify why the different statistics are useful. The recall rate is higher than the precision here because the Random Forest algorithm guesses *too often* that a document is restrained. The result is that it labels a large portion of the human coder's restrained positions correctly. But it also includes several documents that humans label differently.

Depending on the application, scholars may conclude that the supervised method is able to sufficiently replicate human coders. Or, additional steps can be taken to improve accuracy, including: applying other methods, using ensembles of methods, or switching the quantity of interest to proportions.[3] Given the limited space for this demonstration, after validation we move forward and apply the Random Forest classifier to the full set of civilian and political elite statements. The results are presented in Table 3. Stewart and Zhukov (2009) use similar results to show that Russian military elites are actually *more* activist in considering the use of force than their political counterparts, in contrast to the conventional wisdom (Gelpi and Feaver 2002).

6 Discovering Categories and Topics

Supervised and dictionary methods assume a well-defined set of categories. In some instances this poses no real challenge: researchers have a set of categories in mind before collecting texts, either from prior scholarship or a set of hypotheses that form the core of a research project. In other instances, however, the set of categories may be difficult to derive beforehand. For example scholars may struggle to identify the relevant topics of discussion in Senate floor speeches in 1887 or the subject of the daily briefings on the Falklands War.

[3]We provide suggestions for improving supervised learning performance in the Supplementary Material.

This difficulty is due in part to the massive number of potential organizations of even a small number of texts. Consider, for example, the number of ways to partition a set of objects—divide objects into a set of mutually exclusive and exhaustive groups. For any one collection of texts, the set of all possible partitions contains all possible substantively interesting ways to organize the texts (along with many uninteresting ways to organize texts). But enumerating these partitions is impossible. For even moderately sized data sets, say one hundred documents, the number of potential partitions is much larger than the estimated number of atoms in the universe.

Unsupervised learning methods are a class of methods that learn underlying features of text without explicitly imposing categories of interest. Rather than requiring users to condition on known categories beforehand—supervising the methods—unsupervised learning methods use modeling assumptions and properties of the texts to estimate a set of categories and simultaneously assign documents (or parts of documents) to those categories.

Unsupervised methods are valuable because they can identify organizations of text that are theoretically useful, but perhaps understudied or previously unknown. We divide unsupervised categorization methods into two broad categories (Grimmer and King 2011). The most widely used are FAC methods: methods that return a single *clustering* of the input objects. FAC methods are useful, but are *necessarily* model dependent.

Completely resolving the necessary model dependence in unsupervised models is impossible. But two strategies may prove useful at including additional information to make the models more flexible. The first strategy generalizes FAC models, using recently developed statistical models to incorporate context-specific structure into the analysis through a model. Including this information often leads to more interesting clusterings, but necessarily relies on small variations of similar models. The second strategy, CAC, allows researchers to efficiently search over millions of potential categorization schemes to identify interesting or useful organizations of the text. This embraces unsupervised learning methods as a method for generating new categorization schemes, but requires extensive additional analysis to classify all texts into categories.

Regardless of the strategy used to create clusterings, we should still view the output of the clustering methods with skepticism. All the clustering methods are based on incorrect models of language and *a priori* it is hard to know if any one method will provide substantively useful clusterings. Before using a clustering, validating the clustering is essential for demonstrating that the output of an unsupervised method is useful.

The need to validate clusterings does not negate the value of unsupervised methods, nor does it lead to them becoming a special case of supervised learning methods (as suggested in Hillard, Purpura, and Wilkerson 2008). As we show in Section 6.4, validations are done *conditional* on the clustering produced: without first seeing the clustering, assessing validity is impossible. The new organization scheme and documents classified according to those categories is the contribution of the method, not the time difference between applying supervised and unsupervised methods.

In fact, a recent debate in political science casts unsupervised and supervised as competitor methods (e.g., Hillard, Purpura, and Wilkerson 2008; Quinn et al. 2010). This debate is misplaced: supervised and unsupervised methods are different models with different objectives. If there are predetermined categories and documents that need to be placed in those categories, then use a supervised learning method. Using an unsupervised learning method to accomplish this task is at best frustrating—particularly if the predetermined categories are intended to capture subtle differences in tone or sentiment. If, however, researchers approach a problem without a predetermined categorization scheme, unsupervised methods can be useful. But supervised methods will never contribute a new coding scheme.

Far from competitors, supervised and unsupervised methods are most productively viewed as complementary methods, particularly for new projects or recently collected data sets. The categories of interest in a new project or a new corpus are usually unclear or could benefit from extensive exploration of the data. In this case, unsupervised methods provide insights into classifications that would be difficult (or impossible) to obtain without guidance. Once the unsupervised method is fit, we show below how supervised learning methods can be used to validate or generalize the findings.

6.1 *FAC*

We begin with FAC. This class of methods appears in an extensive literature across numerous fields, where they are used to estimate categories from data (for a review, see Manning, Raghavan, and Schütze 2008 and Jain, Murty, and Flynn 1999). We consider two broad classes of FAC models: single membership and mixed membership models.

6.1.1 Single membership models

Single membership clustering models estimate a *clustering*: a partition of documents into mutually exclusive and exhaustive groups. Each group of documents in a clustering is a *cluster*, which represents an estimate of a *category*.[4] Across models, C_i will represent each document's cluster assignment and $\boldsymbol{C} = (C_1, C_2, \ldots, C_N)$ will represent a *partition* (clustering) of documents.

The FAC literature is *massive*, but each algorithm has three components: (1) a definition of document similarity or distance; (2) an objective function that operationalizes an *ideal* clustering; and (3) an optimization algorithm.

To build intuition, we introduce in detail the K-Means algorithm—perhaps the most widely used FAC method (MacQueen 1967). The goal of the K-means algorithm is to identify a partition of the documents that minimizes the squared Euclidean distance within clusters. To obtain this goal, the algorithm produces two quantities of interest: (1) a partition of the documents into K clusters ($k = 1, \ldots, K$) and (2) K cluster centers μ_k. We now describe the three components of the K-means algorithm.

Distance: Standard K-means assumes the distance of a document $\boldsymbol{W}_i$ from a cluster center μ_k is given by squared Euclidean distance,

$$d(\boldsymbol{W}_i, \boldsymbol{\mu}_k) = \sum_{m=1}^{M} (W_{im} - \mu_{km})^2.$$

Many other distance metrics are possible, and each will lead to different clusterings. Further, different *weights* can be attached within a distance metric. For example, scholars commonly use tf-idf weights within a Euclidean distance metric.

Objective Function: Heuristically, a "good" clustering under K-means is a partition where every document is close to its cluster center. Formally, this can be represented with the objective function,

$$\begin{aligned} f(\boldsymbol{\mu}, \boldsymbol{C}, \boldsymbol{W}) &= \sum_{i=1}^{N} \sum_{k=1}^{K} I(C_i = k) d(\boldsymbol{W}_i, \boldsymbol{\mu}_k) \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} I(C_i = k) \left(\sum_{m=1}^{M} (W_{im} - \mu_{km})^2 \right), \end{aligned} \tag{3}$$

where $I(C_i = k)$ is an indicator function, equal to 1 if its argument is true. Equation (3) measures the quality of a clustering and set of cluster centers: the sum of all document's distance from their corresponding cluster centers.

Optimization Method: For any one data set, a distance metric and objective function identify an optimal partition of the data, $\boldsymbol{C}^*$ and cluster centers $\boldsymbol{\mu}^*$. But directly identifying this optimum is extremely difficult—the K-means' objective function is multimodal and non-continuous. Therefore,

[4]Our review of clustering algorithms is necessarily limited. For example, we make no mention of the distinction between hierarchical and nonhierarchical clustering algorithms (Manning, Raghavan, and Schütze 2008). We avoid this distinction between hierarchical clustering algorithms and clustering algorithms that estimate several partitions. As a result, there is a direct mapping from the output of hierarchical clustering algorithms to the partitions we describe here. We also avoid the distinction between soft and hard clustering. Our experience with soft clustering is that these examples usually assign most of a document to essentially one category, resulting in essentially a hard clustering. We do consider mixed membership models below, another form of soft clustering.

K-means, like many other FAC algorithms, employs an approximate and iterative optimization method. The standard K-means optimization algorithm proceeds in two steps. To begin, suppose that we have an initialized set of cluster centers $\boldsymbol{\mu}^{t-1}$. The first step updates each document's assigned cluster to the closest cluster center,

$$C_i^t = \arg\min_k \sum_{m=1}^{M} (W_{im} - \mu_{km})^2.$$

Using the new cluster assignments, $\boldsymbol{C}^t$, each cluster center μ_k is updated by setting it equal to the *average* document assigned to the cluster

$$\boldsymbol{\mu}_k^t = \frac{\sum_{i=1}^{N} I(C_i^t = k)\boldsymbol{W}_i}{\sum_{i=1}^{N} I(C_i^t = k)}.$$

The steps are repeated until the change in objective function falls below a predetermined threshold. This algorithm for optimization—closely related to the Expectation–Maximization algorithm—only guarantees convergence to a local minimum (Dempster, Laird, and Rubin 1977; Bishop 2006) and therefore several random restarts are often necessary to find the global optimum. The K-means algorithm is just one of many potential algorithms for generating partitions of data. The literature is replete with alternative clustering methods. Some of the models vary the distance metric (Manning, Raghavan, and Schütze 2008), suggest different objective functions (Ng, Jordan, and Weiss 2001), and even numerous different optimization procedures (Frey and Dueck 2007). Some of the models do this explicitly with an algorithmic focus, while others vary the three components of the model implicitly in a statistical model for generating new clusterings. See Grimmer and King (2011) and Manning, Raghavan, and Schütze (2008) for more details on the diverse set of methods. While each of the clustering methods are well specified and based on careful derivation, each method relies on a set of assumptions that result in varying performance in different contexts. Unfortunately, the vast literature on clustering algorithms provides little guidance on when specific measures of similarity, objective functions, or optimization algorithms are likely to lead to useful or insightful partitions of data. Indeed, it appears that providing this guidance without specific knowledge of the data set may be impossible (Grimmer and King 2011). Therefore, we think any new clustering methods developed in other fields should be imported into political science with caution. Regardless of the claims made, substance-based validations, as outlined in Section 6.4, are necessary to establish the utility of an unsupervised method.

6.1.2 Mixed membership models

One way to improve the output of single-membership models is to include additional and problem-specific structure. *Topic models* have recently been proposed as one method for including this structure (Blei, Ng, and Jordan 2003). Topic models are a broad class of Bayesian generative models that encode problem-specific structure into an estimation of categories (see Blei 2012 for a review).

Topic models share two broad characteristics. The first is a definition of a topic. Statistically, a *topic* is a probability mass function over words. For a topic k ($k = 1, \ldots, K$), we represent this probability distribution over words with an $M \times 1$ vector $\boldsymbol{\theta}_k$ where θ_{mk} describes the probability the *k-th* topic uses the *m-th* word. Substantively, topics are distinct concepts. In congressional speech, one topic may convey attention to America's involvement in Afghanistan, with a high probability attached to words like `troop`, `war`, `taliban`, and `Afghanistan`. A second topic may discuss the health-care debate, regularly using words like `health`, `care`, `reform`, and `insurance`. To estimate a topic, the models use the co-occurrence of words across documents.

Second, the models share a basic hierarchical structure. The top of the model contains a prior which borrows information across units. In the middle of the structure is a *mixed membership*

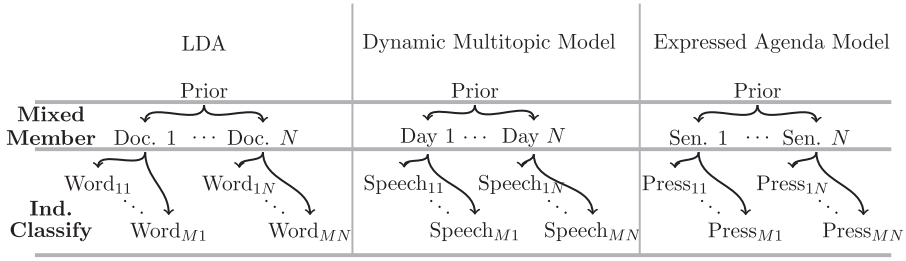

Fig. 2 A common structure across seemingly unrelated text models: in each case, the mixed member allows the scholar to estimate the parameter of particular substantive interest. The prior facilitates the borrowing of information across units, and the individual elements classified form the observed data.

level: this measures how a unit of interest allocates its attention to the estimated topics. And at the bottom of the hierarchy a word or document is assigned to a *single* topic.

The first and most widely used topic model, latent Dirichlet allocation (LDA), is one example of how this structure is used in a statistical model (Blei, Ng, and Jordan 2003). The left-hand side of Fig. 2 provides a nontechnical overview of LDA's data generation process.

LDA assumes that each *document* is a mixture of topics. For each document, i represent the proportion of the document dedicated to topic k as π_{ik} and collect the proportions across topics to be $\boldsymbol{\pi}_i = (\pi_{i1}, \pi_{i2}, \ldots, \pi_{iK})$. We will suppose that each document's proportions are drawn from a common Dirichlet prior,

$$\pi_i \sim \text{Dirichlet}(\alpha)$$

where α represent the Dirichlet distribution's shape parameters.

Within each document, the words are drawn according to the distribution of topics. Suppose that a document contains N_i total words ($j = 1, \ldots, N_i$). LDA assumes that a two-step process generates each word. To obtain the *j-th* word in the *i-th* document, the first step is to draw its topic τ_{ij},

$$\boldsymbol{\tau}_{ij} \sim \text{Multinomial}(1, \pi_i).$$

Conditional on topic assignment, the actual word is drawn: if the *j-th* word in the *i-th* document is assigned to the *k-th* topic, then we draw from the corresponding topic,

$$W_{ij} \sim \text{Multinomial}(1, \boldsymbol{\theta}_k).$$

Topic Models in Political Science: Political scientists have extended topic models so that the parameters correspond to politically relevant quantities of interest. The first topic model in political science is the dynamic multitopic model (Quinn et al. 2010), used to model the daily attention to topics in Senate floor speeches (center panel of Fig. 2). Following Quinn et al. (2010) is the *expressed agenda model* (Grimmer 2010), which measures the attention senators allocate to press releases (right-hand panel of Fig. 2). There are numerous novel features of both models that are missed when considered abstractly, but we will see that considering the models abstractly leads to a useful realization about the common elements of all three models. The dynamic multitopic model in Quinn et al. (2010) supposes that each day in the Senate is a mixture of attention to topics: days are at the mixed membership level. Each speech is then assigned to a single topic, analogous to assigning individual words to topics in LDA. And finally, a dynamic prior is used to make more efficient inferences about the proportion of each day's speeches allocated to each topic. The center panel of Fig. 2 shows the similarity of this model with LDA. The expressed agenda model in Grimmer (2010) demonstrates yet another way to exploit the same structure. The expressed agenda model is designed to measure how authors divide attention over topics and is applied to model how senators explain work to constituents in press releases. The key assumption is that each senator divides her attention over the set of topics: senators are at the mixed membership level in the model. Then, conditional on the senator's mixture of attention to topics, the topic of each press release is drawn.

The right-hand panel of Fig. 2 shows the similarity with LDA and the dynamic topic model. Like days in the dynamic topic model and documents in LDA, senators in the expressed agenda model are allowed to be members of several topics. And analogous to speeches in the dynamic topic model and words in LDA, each press release is assigned to a single topic and the prior is used to make more efficient inferences.

The similarity in structure across the three models in Fig. 2 demonstrates how statistical topic models can facilitate the production of substantively interesting and new models. Situating the models in a data-generating process makes including additional structure straightforward. For example, we could combine the dynamic topic model and the expressed agenda model to model how senators' attention to topics vary over time. More generally, we could include other problem-specific information, like the order of speeches made in the Senate in any one day, the sequence of ideas exchanged during a press conference, or covariates about the authors of texts. But requiring political scientists to tune each model to their task is a daunting task. The result is that only a few models are likely to be used for analyzing texts. Further, while the statistical structure allows for the inclusion of problem-specific information, the model is still limited to a single family of probability distributions and therefore a specific set of assumptions. CAC methods provide a way to explore these different assumptions.

6.2 *CAC*

Identifying the assumptions that will lead to a useful clustering of documents in a particular data set beforehand is difficult. After the fact, however, it is much easier to assess whether an organization of texts is useful within a particular context. With this idea in mind, CAC is a method for efficiently searching over a large space of clusterings (Grimmer and King 2011). To start, a diverse set of FAC methods is applied to a data set. The different methods vary the definition of similarity, objective functions, and optimization algorithms to provide diverse ways to organize the documents. Then, Grimmer and King (2011) show how to embed the partitions into a two-dimensional space such that two clusterings are close in the space if they organize the documents in similar ways. Using this space, Grimmer and King (2011) introduce a method for exploring it, easily searching over the included methods and millions of other clusterings that are the result of combinations of similar organizations of the data.

CAC has three broad uses. First, the method provides a way to identify new or understudied concepts—even in already extensively analyzed texts. The diverse set of partitions included ensures that researchers will encounter new ways of organizing data. Second, CAC provides an accelerated way to explore new collections of texts. CAC ensures that these explorations are not limited to only one organization of the data. Third, CAC provides a way to evaluate the originality of new clustering methods. If a clustering method is truly new, then it should occupy a distinct part of the space—at least for some collections. If it is useful, then the distinct clusterings should lead to new or useful organizations of the documents that other methods miss.

But CAC does place substantial burden on users to identify clusterings that are useful. In one respect, this is a virtue of the method—it is limited only by human creativity. But, it does imply that the method will only perform as well as the person conducting the exploration and interrogation of the space.

6.3 *Setting the Number of Clusters*

Both CAC and FAC methods require setting the number of clusters in a model. For K-means the number of clusters—K—has to be set, for mixed membership models the number of topics must be chosen, and for CAC the number of clusters in the final clustering must be determined. Determining the number of clusters is one of the most difficult questions in unsupervised learning. Some methods attempt to eliminate this decision and estimate the number of features (Frey and Dueck 2007; Wallach et al. 2010), but recent studies show that the estimated number of clusters is strongly model dependent (Wallach et al. 2010). We also cannot turn to fit statistics, as Chang et al. (2009) show

that there is often a negative relationship between the best-fitting model and the substantive information provided.

When setting the number of clusters, we caution in general that *you can't get something for nothing*. Models that estimate the number of clusters are heavily model dependent (Wallach et al. 2010). Nonparametric models, such as the Dirichlet process prior, still make model-based decisions on the number of clusters or topics to include. But the choice has been reparameterized as a hyper prior (Wallach et al. 2010) or as a tuning parameter in an algorithm (Frey and Dueck 2007).

Rather than statistical fit, model selection should be recast as a problem of measuring *substantive fit*. Unsupervised methods for content analysis *reduce* the information in large text collections substantially. Measures of statistical fit measure how well the models fit by comparing the estimated parameters to the actual data. But this relies on the assumption that the goal is to model well the representation of texts after preprocessing. It is not. The preprocessed texts represent a substantial simplification of the documents. The goal is revelation of substantively interesting information.

We think a productive line of inquiry will replace the use of the preprocessed texts with carefully elicited evaluations based on the substance of the model. Quinn et al. (2010) provide one method for performing this model selection. At the first stage of the process, candidate models are fit varying the number of clusters or topics. At the second stage, human judgment is used to select a final model, assessing the *quality* of the clusters. This can be done approximately—assessing whether the clusters group together documents that are distinct from other clusters and internally consistent. Or an explicit search across models based on elicited subject expert evaluations can be employed, using measures developed in Chang et al. (2009) or Grimmer and King (2011).

6.4 *Validating Unsupervised Methods: How Legislators Present Their Work to Constituents*

As Quinn et al. (2010) observe, unsupervised methods shift the user burden from determining categories before analysis to validating model output afterward. The post-fit validations necessary can be extensive, but the methods are still *useful* because they suggest new, or at least understudied, ways to organize the data. As an example of this process, we perform a validation of senators' expressed priorities introduced in Grimmer (2012). To obtain the measures of expressed priorities, Grimmer (2012) applies a version of the expressed agenda model to over 64,000 Senate press releases issued from 2005 to 2007.

Applying the unsupervised model leads to an understudied organization of senators, based on how they present their work to constituents. Grimmer (2012) shows that the primary variation underlying senators' expressed priorities is how senators balance position taking and credit claiming in their press releases. Some senators allocate substantial attention to articulating positions, others allocate much more attention to credit claiming, and still others adopt a more evenly balanced presentational style. Grimmer (2012) shows that this spectrum is theoretically interesting: it approximates spectra suggested in earlier qualitative work (Fenno 1978; Yiannakis 1982), and predicted in formal theoretic models of Congress and Congressional elections (Weingast, Shepsle, and Johnsen 1981; Ashworth and Bueno de Mesquita 2006). And Grimmer (2012) shows that where senators fall on this spectrum has real consequences for representation.

To make these inferences, however, requires extensive validation of both the estimated topics and expressed priorities. We overview some of those validations now—of both the topics and the expressed priorities. Before proceeding, we caution that the validations performed here are only a subset of the evaluations any researcher would need to perform before using measures from unsupervised methods in their own work. For a more comprehensive review of validity and topic models, see Quinn et al. (2010).

6.4.1 Validating topics

As a prelude to validating the topic output, the topics must be labeled: it must be determined *what* each topic measures. Table 4 provides three examples of the topics from Grimmer (2012), based on the broader forty-four topic model. One method for labeling is reading: sampling ten to fifteen documents assigned to a topic and inferring the commonality across the press releases. Examples of

Table 4 An example of topic labeling

Description	*Discriminating words*
Iraq War	Iraq, iraqi, troop, war, sectarian
Honorary	Honor, prayer, remember, fund, tribute
Fire Department Grants	Firefight, homeland, afgp, award, equipment

the labels are posted in the first column of Table 4. Statistical methods are also used. While many have been proposed, they share the same intuition: identify words that are highly predictive of documents belonging to a particular topic. Examples of the *discriminating* words are also in Table 4

The topics in Table 4 exemplify the utility of unsupervised learning. The first row of Table 4 identifies a topic about the Iraq war—one of the most salient debates during the time the press releases were issued. But the model also identifies a topic of press releases that *honor* constituents—commemorating a national holiday or a tribute to a deceased constituent. The final row of Table 4 is a topic about claiming credit for grants allocated to fire departments through the Assistance to Firefighter Grant Program (AFGP). This is a prominent type of credit claiming—bureaucratic agencies creating opportunities for legislators—often missed from standard models of legislative speech.

With the topic labels in hand for all forty-four topics, we now describe two methods for validating the topic labels. The first evaluation assesses *semantic validity* (Quinn et al. 2010): the extent to which our topics identify coherent groups of press releases that are internally homogenous, yet distinctive from other topics. The second evaluation assesses *predictive validity* (Quinn et al. 2010): measuring how well variation in topic usage corresponds with expected events.

Semantic Validation via Experiments: To assess *semantic* validity, Grimmer and King (2011) introduce a method based on experimentally elicited human input. Using this method allows direct comparison of the quality of clusterings, but it is unable to provide an absolute measure of semantic validity. To assess the semantic validity of the topics, we compare them to two clusterings produced by Senate press secretaries and a clustering from a state-of-the-art nonparametric clustering method (Blei and Jordan 2006). Following Grimmer and King (2011), we first sample pairs of documents assigned to the same and different clusters. Research assistants were then asked to evaluate the pairs of documents on a three-point scale, rating a pair of press releases as (1) unrelated, (2) loosely related, or (3) closely related. We then average over the human evaluations to create the measure of cluster quality: the average evaluation of press releases assigned to the same cluster and less the average evaluation of press releases assigned to different clusters. For a method i call this $Cluster\ Quality_i$. We then compare alternative partitions to the clustering used in Grimmer (2012) to compare relative semantic validity, $\text{Cluster Quality}_{\text{ExpressedAgenda}} - \text{Cluster Quality}_{\text{Alt.Method}}$. The differences are plotted in Fig. 3, with each point representing the mean difference and the thick and thin lines constituting 80% and 95% credible intervals for the difference, respectively.

Figure 3 shows that the expressed agenda model applied to the press releases produces a higher quality clustering—more semantically valid—than any of the comparison methods on the same press releases. This includes clusterings from the nonparametric method and clusterings produced by two Senate press secretaries.

Predictive Validity: Quinn et al. (2010) argue that if topics are valid, then external events should explain sudden increases in attention to a topic (Grimmer 2010 performs a similar validation). Figure 4 plots the number of press releases issued each day about the Iraq war (top plot) and honorary press releases (bottom plot), with the date plotted on the vertical axis.

Both plots in Fig. 4 show that external events predict spikes in attention to press releases. When major events happen surrounding the war—such as the release of the Iraq study group report—more press releases are issued about Iraq. Likewise, when major figures die, such as Pope John Paul II, more honorary press releases are issued.

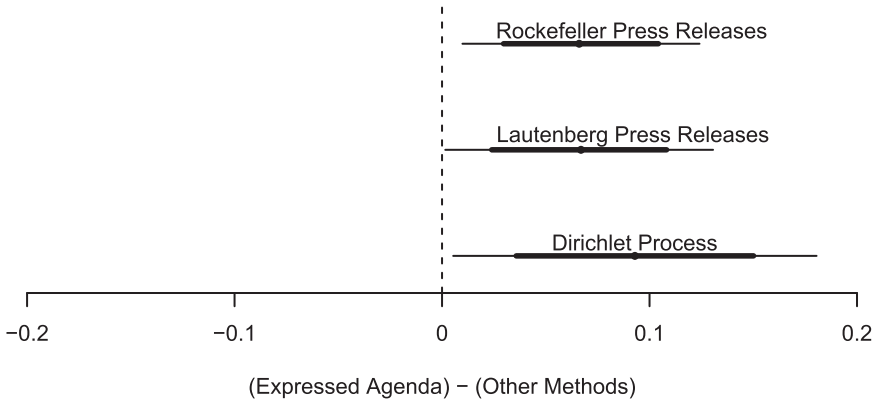

Fig. 3 Semantic validity of topics. This figure shows that the model applied in Grimmer (2012) provides higher quality clusters than press secretaries grouping press releases on senators' Web sites and state-of-the-art nonparametric topic models.

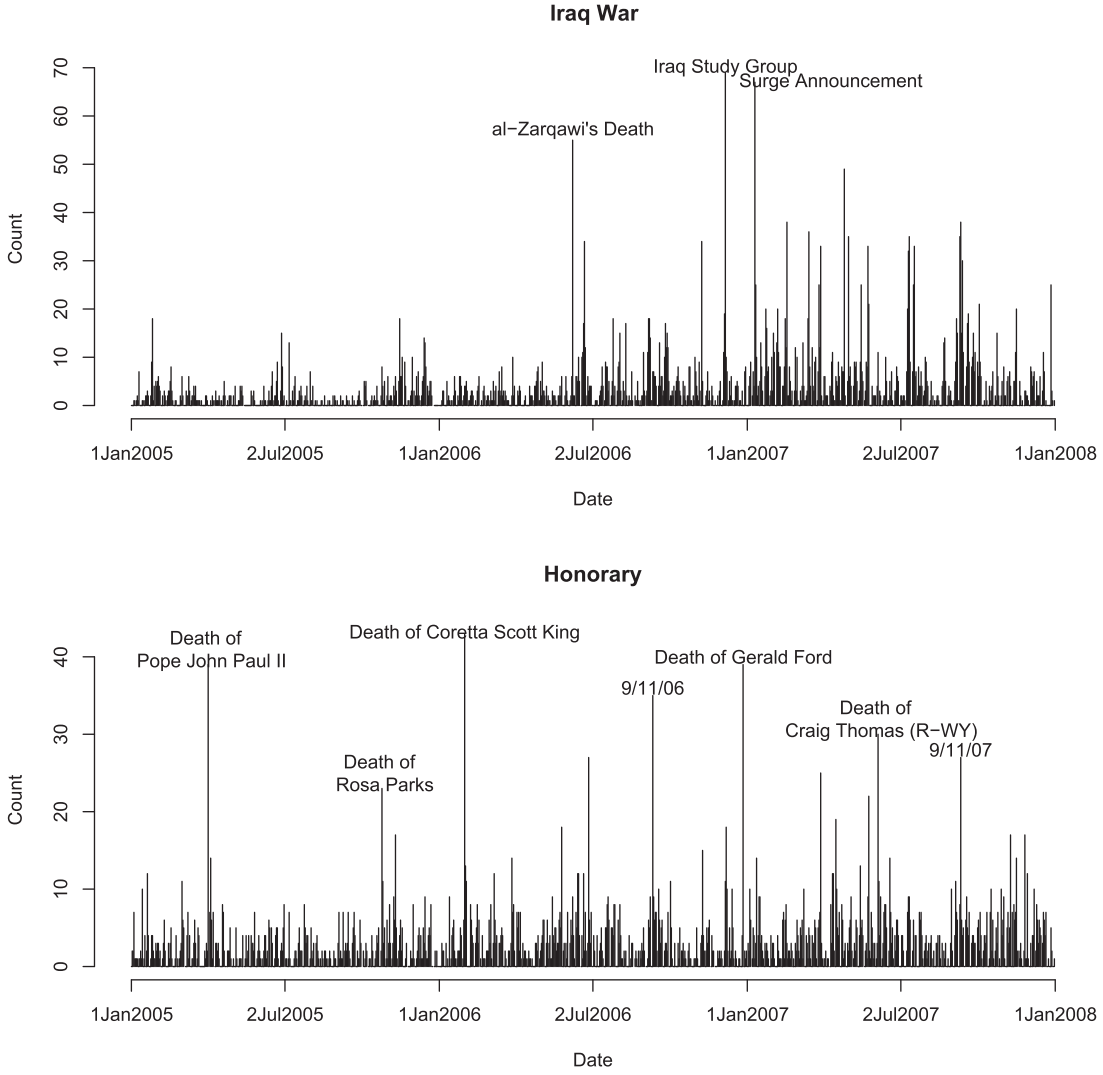

Fig. 4 Predictive validity of topics.

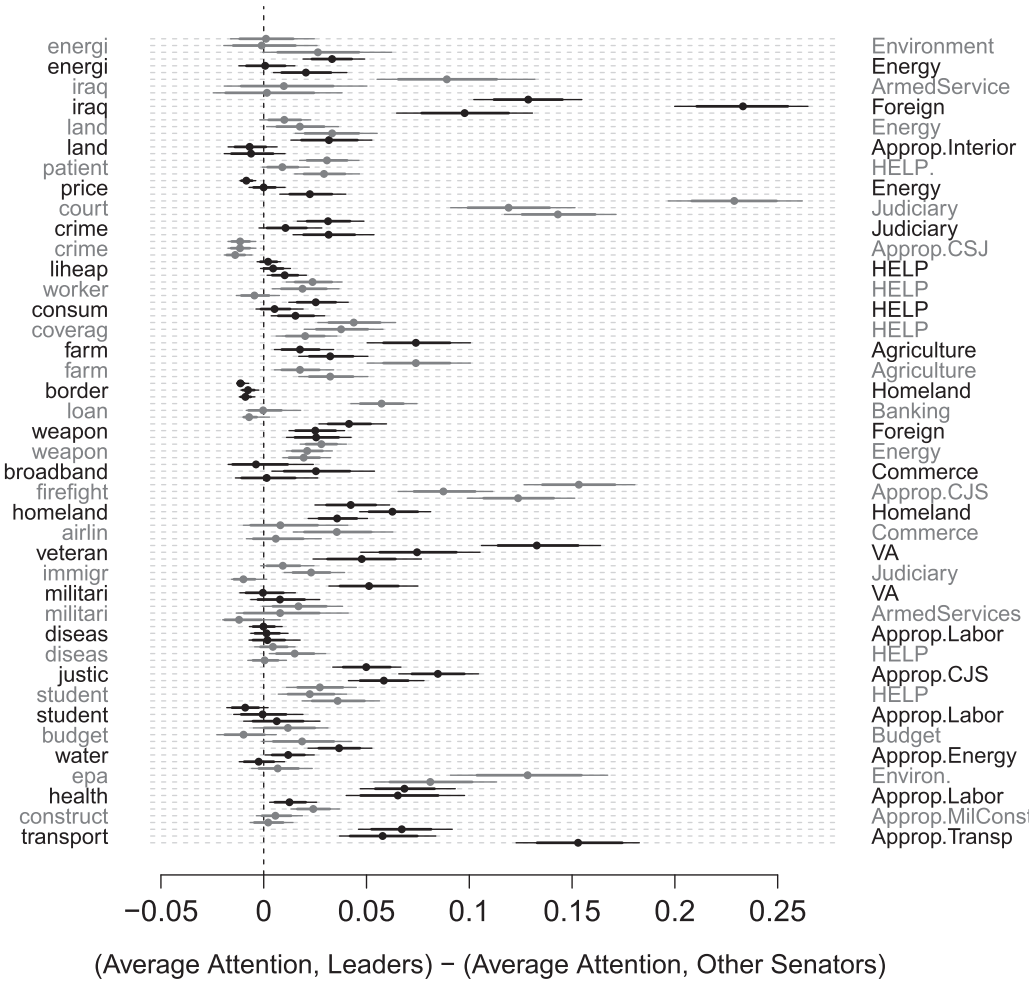

Fig. 5 Predictive validity of expressed priorities. This figure compares the attention that Senate committee leaders—chairs or ranking members—dedicate to topics under their committee jurisdictions to the attention allocated by the rest of the Senate. The solid black dots represent the expected difference, the thick lines are 80% credible intervals, and the thin lines are 95% intervals. Along the left-hand vertical axis, the topics are listed, and, on the right-hand side, the corresponding committee names are listed. As this figure clearly illustrates, committee leaders allocate substantially more attention to issues under their jurisdiction than other members of Congress.

6.4.2 Validating expressed priorities

The expressed agenda model also produces a measure of senators' expressed priorities: how senators divide their attention over the topics. This too requires rigorous validation—a demonstration that the estimated priorities measure the concept claimed. We apply two tests to the expressed priorities from Grimmer (2012). First, we show that they satisfy one test of *predictive validity*. Second, we show that the measures have convergent validity with supervised learning methods created after observing the categorization scheme from the unsupervised method.

Predictive Validity: Grimmer (2010) argues that the leaders of Senate committees—committee chairs and ranking members—should allocate more attention to issues that fall under the jurisdiction of their committee than other senators. This straightforward expectation provides a test of the predictive validity of the estimated priorities. Figure 5 carries out this comparison. In Fig. 5, committee leaders' average attention dedicated to an issue under their committee's jurisdiction is compared with the average attention among the other ninety-eight senators for forty committee topic pairs, for each year in the analysis. The left-hand vertical axis denotes the topics that were

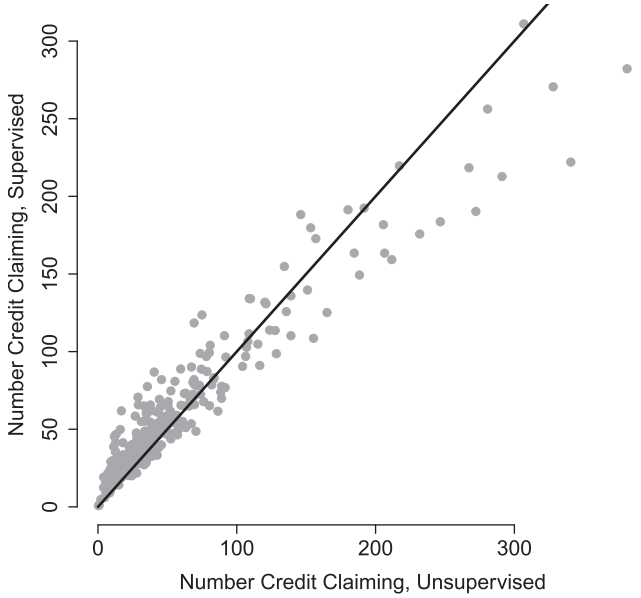

Fig. 6 Convergent validity of unsupervised methods with supervised methods.

used for the comparison and the right-hand vertical axis contains an abbreviated committee or appropriations subcommittee name. The solid dot represents the expected difference between committee leaders and the rest of the Senate; the thick lines are 80% and 95% credible intervals, respectively. If committee leaders discuss issues related to their committee more often, then the estimates should be to the right of the vertical dotted line at zero.

Figure 5 shows that committee leaders allocate more attention to issues under their committee's jurisdiction than the average senator. For almost every topic committee pair in each year, leaders of Senate committees allocate substantially more attention to issues under their jurisdiction than other senators. This suggests that the measures are, at least by this test, predictively valid.

Convergent Validity: Even after several validations, skepticism can remain about measures from an unsupervised method. This skepticism is perhaps most acute in political science, where unsupervised methods have a colored past (Armstrong 1967). This skepticism would be allayed—at least partially—if we were confident that an unsupervised measure was just as valid as the equivalent measure from a supervised method. Here, we provide one way to gain this confidence by using supervised methods to validate the output of unsupervised methods. For simplicity and space, we perform this validation on a subset of the full expressed priorities used in Grimmer (2012). Grimmer (2012) shows that several of the topics identify press releases that are claiming credit for money directed to the district (Mayhew 1974) (the differences across topics capture differences in the type of money claimed). Grimmer (2012) aggregates across categories to create an estimate of the number of press releases senators issue claiming credit for money in their state. We validate this measure using ReadMe. First, we developed a codebook that contained a category for claiming credit for money, along with other categories to improve our performance (Mayhew 1974). We then asked a research assistant to classify five hundred press releases according to our scheme. Then, we used those five hundred press releases and ReadMe to measure the number of credit-claiming press releases issued.[5] Figure 6 shows the strong correlation between the supervised and unsupervised methods. In it, each senator's estimated number of credit-claiming press releases are plotted against the estimate from the expressed agenda model. The black line in the plot is the 45° line: if the two estimates were equivalent, the points would fall along this line.

[5] `ReadMe` estimates proportions, so the number of press releases in each category can be retrieved by multiplying by the total number of press releases.

Figure 6 shows clearly that the gray points group around the 45° line, demonstrating that the unsupervised method is essentially equivalent to the supervised method. This is true across all senators, resulting in a correlation of 0.96. The unsupervised estimate of credit-claiming behavior is essentially *equivalent* to the estimate from ReadMe. This implies that all the confidence we would attach to using the estimates from the supervised method can also be attached to the unsupervised method. We caution that this validation *does not* obviate the need for unsupervised methods. The validation using the supervised method is possible *only after the unsupervised method suggests a classification scheme*. It does, however, provide one direct test to ensure that the output from an unsupervised method is just as valid, reliable, and useful as the categorization schemes from supervised methods. It also serves to underscore the point that future work based on a categorization scheme developed by an unsupervised method may well use supervised classification to extend and generalize that point.

7 Measuring Latent Features in Texts: Scaling Political Actors

One of the most promising applications of automated content analysis methods is to locate political actors in ideological space. Estimating locations using existing data is often difficult and sometimes impossible. Roll call votes are regularly used to scale legislators (Poole and Rosenthal 1997; Clinton, Jackman, and Rivers 2004), but outside the U.S. Congress roll call votes are less reliable (Spirling and McLean 2007). And other political actors—presidents, bureaucrats, and political candidates—do not cast votes. Other methods for scaling political actors have been developed (e.g., Gerber and Lewis 2004; Bonica 2011), but they rely on particular disclosure institutions that are often absent in other democracies.

But nearly all political actors speak. A method that could use this text to place actors in a political space would facilitate testing some of the most important theories of politics. We describe two methods for scaling political actors using texts. One method, based on Laver, Benoit, and Garry (2003), is a supervised method—analogous to dictionary methods—to situate actors in space based on their words. A second method is an unsupervised method for locating actors in space (Monroe and Maeda 2004; Slapin and Proksch 2008).

The scaling literature holds great promise for testing spatial theories of politics. Recognizing this, several recent papers have offered important technical contributions that improve the methods used to perform the scalings (Martin and Vanberg 2007; Lowe 2008; Lowe et al. 2011). These papers are important, but we think that the scaling literature would benefit from a clearer articulation of its goals. Recent papers have implicitly equated the goal of scaling methods as replicating expert opinion (Benoit, Laver, and Mikhaylov 2009; Mikhaylov, Laver, and Benoit 2010) or well-validated scalings made using nontext data (Beauchamp 2011). Certainly plausibility of measures is important, but if the goal is to replicate expert opinion, or already existent scalings, then text methods are unnecessary. Simple extrapolation from the experts or existing scaling would suffice.

One clear goal for the scaling literature could be *prediction* of political events. Beauchamp (2011), for example, shows that the output from text scaling methods can be used to predict votes in Congress. More generally, text scalings should be able to predict legislative coalitions throughout the policy creation process. Or when applied to campaigns, text scalings should be able to predict endorsements and campaign donations.

Improving the validation of scales will help improve current models, which rely on the strong assumption of *ideological dominance* in speech. Both supervised and unsupervised scaling methods rely on the strong assumption that actors' ideological leanings determine what is discussed in texts. This assumption is often useful. For example, Beauchamp (2011) shows that this works well in Senate floor speeches, and we replicate an example from Slapin and Proksch (2008) that shows that the model works well with German political platforms. But in other political speech, this may not be true—we show below that the ideological dominance assumption appears to not hold in Senate press releases, where senators regularly engage in nonideological credit claiming.

Scaling methods will have more even performance across texts if they are accompanied with methods that separate ideological and non-ideological statements. Some of this separation is now done manually. For example, it is recommended in Slapin and Proksch (2008). But more nuanced methods for separating ideological content remain an important subject of future research.

7.1 *Supervised Methods for Scaling*

Laver, Benoit, and Garry (2003) represent a true breakthrough in placing political actors in a policy space. Rather than rely on difficult to replicate and hard to validate manual coding or dictionary methods, Laver, Benoit, and Garry (2003) introduced a fully automated method for scaling political actors: *wordscores*.

Wordscores is a special case of the dictionary methods that we presented in Section 5.1. The first step is the selection of *reference* texts that define the political positions in the space. In the simplest example, we may select two texts to define the liberal and conservative ends of the spectrum. If we wanted to scale U.S. senators based on their speeches, for example, we may define as a reference text all the speeches from a very liberal senator, like Ron Wyden (D-OR) or Barbara Boxer (D-CA), and a very conservative senator, like Tom Coburn (R-OK) or Jim DeMint (R-OK). The reference (training) texts are then used to generate a *score* for each word. The score measures the relative rate each word is used in the reference texts. This creates a measure of how well the word separates liberal and conservative members—one measure of whether a word is *liberal* or *conservative*. The word scores are then used to scale the remaining texts. Laver, Benoit, and Garry (2003) calls these the *virgin* texts, but in supervised learning we would call these texts the *test set*. To scale the documents using the word scores, first Laver, Benoit, and Garry (2003) calculate the relative rate words are used in each of the test documents. The position of the texts is then determined by taking the weighted average of the word scores of the words in a text, where the weights are given by the rate at which the words are used.

Wordscores is rich and generalizable to multiple dimensions and to include several reference texts. But facets of wordscores constrain the method and make it difficult to recommend for general use (see Lowe 2008 for an extended critique). By defining "liberal" and "conservative" using *only* reference texts, Laver, Benoit, and Garry (2003) conflate ideological language with stylistic differences across authors and impose the ideological dominance assumption on the texts. The result is that every use of wordscores will depend strongly on the reference texts that are used, in part because of stylistic differences across authors and in part because the reference texts will discuss non-ideological content. Careful preprocessing of texts, to remove words that are likely only stylistic, can mitigate part of this problem. Beauchamp (2011), for instance, shows that results are significantly improved by removing technical language, which coincides more with party power than with ideology. But no amount of preprocessing can completely eliminate it. The explicit adoption of a supervised learning approach might limit the influence of style substantially. Unfortunately, this also requires a substantial increase in effort and time, which makes it application unwieldy.

7.2 *Unsupervised Methods for Scaling*

Rather than rely on reference texts for scaling documents, unsupervised scaling methods *discover* words that distinguish locations on a political spectrum. First Monroe and Maeda (2004), then later Slapin and Proksch (2008), introduce statistical models based on *item response theory* (IRT) to automatically estimate the spatial location of the parties. Politicians are assumed to reside in a low-dimensional political space, which is represented by the parameter θ_i for politician i. A politician's (or party's) position in this space is assumed to affect the rate at which words are used in texts. Using this assumption and text data, unsupervised scaling methods estimate the underlying positions of political actors.

Slapin and Proksch (2008) develop their method, *wordfish*, as a Poisson-IRT model. Specifically, Slapin and Proksch (2008) assume that each word j from individual i, W_{ij} is drawn from a Poisson distribution with rate λ_{ij}, $W_{ij} \sim \text{Poisson}(\lambda_{ij})$. λ_{ij} is modeled as a function of individual i's

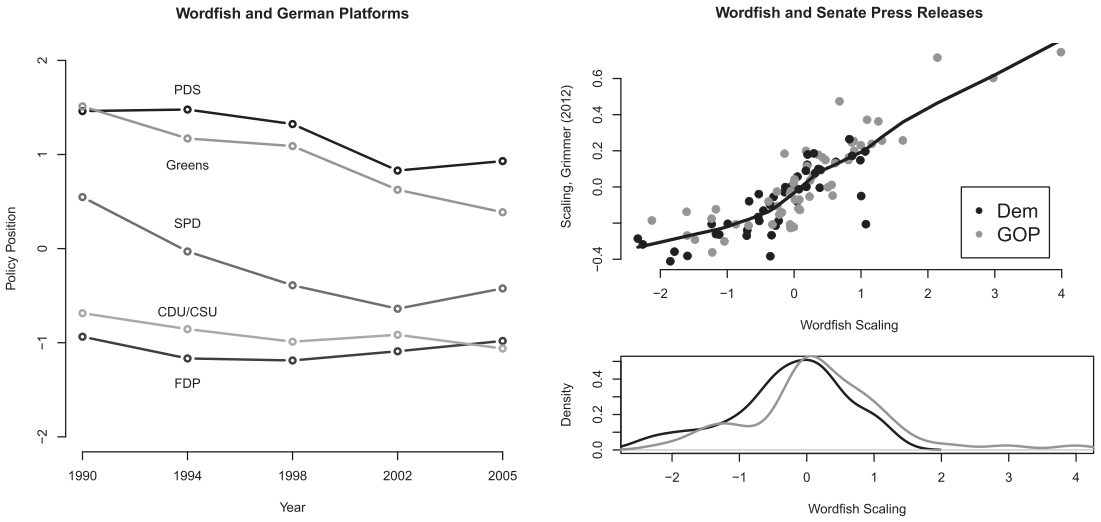

Fig. 7 Wordfish algorithm: performance varies across context.

loquaciousness (α_i), the frequency word j is used (ψ_j), the extent to which a word discriminates the underlying ideological space (β_j), and the politician's underlying position (θ_i),

$$\lambda_{ij} = \exp(\alpha_i + \psi_j + \beta_j \times \theta_i).$$

The next section applies this model to political texts from two different contexts, demonstrating conditions when the model is able to reliably retrieve underlying policy positions.

7.3 *Applying Unsupervised Methods to Political Texts*

Both the strength and limitations of IRT methods for scaling are the lack of supervision. When the model works well it provides reliable estimates of political actors' spatial locations with little resource investment. But the lack of supervision and the use of an IRT model implies that the model will seize upon the primary variation in language across actors. This might be ideological. Or, the differences across actors may be about their focus on policy or pork, the style in which the essays were written, or the tone of the statements. Because the model does not include supervision explicitly, it is difficult to guarantee that the output of the model will reliably identify the revealed ideological locations of political actors. It is worth emphasizing that this *is not* a shortcoming of wordfish. In fact, we will show below that non-ideological locations that wordfish identifies are quite useful. But this does suggest that one should not assume that wordfish output measures an ideological location without careful validation.

When the ideological dominance assumption fits the data the model can reliably retrieve valid ideological dimensions from political texts. Take, for example, the left-hand plot of Fig. 7. This replicates a plot in Fig. 1 of Slapin and Proksch (2008), who apply the wordfish algorithm to German party platforms. As Slapin and Proksch (2008) show, the estimates in the left-hand plot separate the German parties and replicate expert assessments of German party movement over time.

But when ideological dominance assumption fails to fit the data, wordfish fails to retrieve underlying policy dimensions. The right-hand plot applies the wordfish algorithm to Senate press-release data introduced in Grimmer (2012). The bottom right-hand plot in Fig. 7 is a density of the Democrats (black line) and Republican (gray line) positions from the wordfish algorithm. The model clearly fails to separate Democrat and Republican senators—a necessity for any valid scaling in the now polarized Senate.

The wordfish scaling is meaningful substantively, but it does not correspond to standard policy space. The top plot shows that the wordfish algorithm reproduces the spectrum Grimmer (2012)

identified using the expressed agenda model—how senators balance position taking and credit claiming in press releases. This plot presents the scaling from Grimmer (2012) against the scaling from wordfish on the horizontal axis and the black line is a lowess curve (Cleveland 1979). The relationship between the two measures is extremely strong—correlating at 0.86. Clear evidence that wordfish has identified this interesting—though nonideological—spectrum in the Senate press releases.

This exemplifies when scaling methods are likely to recover an ideological position. When political actors are engaging in heavily ideological speech—as in German party platforms—unsupervised methods appear to retrieve reliable position estimates. But when political actors can avoid ideological speech—as in Senate press releases—scaling methods retrieve some other, nonideological scaling. Therefore when applying scaling algorithms, careful validation is needed to confirm that the intended space has been identified. And an essential future area of future research will simultaneously isolate ideological statements and then employ those ideological statements to scale political actors.

8 Text as Data in Political Science

Automated content analysis methods provide a wide range of tools to measure diverse quantities of interest. This ranges from classifying documents—either into existing or yet to be determined categories—or scaling political actors into policy space. We emphasize that any one method's performance will be context specific. And because text analysis methods are necessarily *incorrect* models of language, the output always necessitates careful validation. For supervised classification methods, this requires demonstrating that the classification from machines replicates hand coding. For unsupervised classification and scaling methods, this requires validating that the measures produced correspond with the concepts claimed.

The automated content literature extends well beyond the methods discussed in this article. Textbooks produced for other fields provide excellent overviews of methods not discussed here, including natural language processing tools. (Manning, Raghavan, and Schütze 2008; Jurafsky and Martin 2009). We also recommend political science papers that make use of other methods not profiled here (e.g., Schrodt 2000).

While there are many possible paths for this research to advance along, we identify three of the most important here.

New Texts Need New Methods: Perhaps the most obvious future research pursuit will be the development of new statistical models for text. Indeed, this is already actively underway within political science, complementing long-standing literatures in computer science, statistics, and machine learning. New text data in political science will necessitate the development of new methods. But as methodologists develop problem-specific tools, they should also think generally about their methods. Identifying commonalities will allow scholars to share creative solutions to common problems.

Uncertainty in Automated Content Methods: Measuring uncertainty in automated text methods remains one of the most important challenges. One of the greatest strengths of the quantitative treatment of text as data is the ability to estimate uncertainty in measurements. And there has been progress in measuring uncertainty, particularly in supervised classification methods. Hopkins and King (2010) show how *simulation-extrapolation* (SIMEX) can allow for uncertainty in the categories human coders place training documents. Similarly, Benoit, Laver, and Mikhaylov (2009) use SIMEX to include error in text-based scales into generalized linear models. But solutions across models are needed. This may take the form of characterizing full posterior distributions for Bayesian statistical models, determining fast and reliable computational methods for algorithmic models, or methods for including uncertainty generated when including humans in the analysis process.

New Frontiers: *New Texts and New Questions*: Beyond methodological innovation, there are vast stacks of texts that can now be analyzed efficiently using automated text analysis. From political theory, to law, to survey research, scholars stand to learn much from the application of automated text analysis methods to their domain of interest. Political scientists may also use texts to

accomplish tasks beyond those that we have highlighted here. Part of the utility of these texts is that they will provide new data to test long-standing theories. But the new texts can also suggest new ideas, concepts, and processes previously undiscovered.

The vast array of potential applications captures well the promise of automated text analysis and its potential pitfalls. The promise is that the methods will make possible inferences that were previously impossible. If political scientists can effectively use large collections of texts in their inferences, then many substantively important questions are likely to be answered. The pitfalls are that applying these methods will take careful thought and reasoning. Applying any one of the methods described here without careful thought will likely lead to few answers and a great deal of frustration.

This essay provides some guidance to avoid these pitfalls. If scholars recognize the limitations of statistical text models and demonstrate the validity of their measurements, automated methods will reach their promise and revolutionize fields of study within political science.

Funding

Brandon Stewart gratefully acknowledges a Graduate Research Fellowship from the National Science Foundation.

References

Adler, E. Scott, and John Wilkerson. 2011. *The Congressional bills project*. http://www.congressionalbills.org.

Ansolabehere, Stephen, and Shanto Iyengar. 1995. *Going negative: How political advertisements shrink and polarize the electorate*. New York, NY: Simon & Schuster.

Armstrong, J. S. 1967. Derivation of theory by means of factor analysis or Tom Swift and his electric factor analysis machine. *The American Statistician* 21(1):17–21.

Ashworth, Scott, and Scott Bueno de Mesquita. 2006. Delivering the goods: Legislative particularism in different electoral and institutional settings. *Journal of Politics* 68(1):168–79.

Beauchamp, Nick. 2011. Using text to scale legislatures with uninformative voting. New York University Mimeo.

Benoit, K., M. Laver, and S. Mikhaylov. 2009. Treating words as data with error: Uncertainty in text statements of policy positions. *American Journal of Political Science* 53(2):495–513.

Berinsky, Adam, Greg Huber, and Gabriel Lenz. 2012. Using mechanical turk as a subject recruitment tool for experimental research. *Political Analysis* 20:351–68.

Bishop, Christopher. 1995. *Neural networks for pattern recognition*. Gloucestershire, UK: Clarendon Press.

———. 2006. *Pattern recognition and machine learning*. New York, NY: Springer.

Blei, David. 2012. Probabilistic topic models. *Communications of the ACM* 55(4):77–84.

Blei, David, Andrew Ng, and Michael Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning and Research* 3:993–1022.

Blei, David, and Michael Jordan. 2006. Variational inference for dirichlet process mixtures. *Journal of Bayesian Analysis* 1(1):121–44.

Bonica, Adam. 2011. Estimating ideological positions of candidates and contributions from campaign finance records. Stanford University Mimeo.

Bradley, M. M., and P. J. Lang. 1999. Affective Norms for English Words (ANEW): Stimuli, instruction, manual and affective ratings. University of Florida Mimeo.

Breiman, L. 2001. Random Forests. *Machine Learning* 45:5–32.

Budge, Ian, and Paul Pennings. 2007. Do they work? Validating computerised word frequency estimates against policy series. *Electoral Studies* 26:121–29.

Burden, Barry, and Joseph Sanberg. 2003. Budget rhetoric in presidential campaigns from 1952 to 2000. *Political Behavior* 25(2):97–118.

Chang, Jonathan, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei 2009. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, eds. Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, 288–96. Cambridge, MA: The MIT Press.

Cleveland, William S. 1979. Robust locally weighted regression and scatterplots. *Journal of the American Statistical Association* 74(368):829–36.

Clinton, Joshua, Simon Jackman, and Douglas Rivers. 2004. The statistical analysis of roll call data. *American Political Science Review* 98(02):355–70.

Dempster, Arthur, Nathan Laird, and Donald Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* 39(1):1–38.

Diermeier, Daniel, Jean-Francois Godbout, Bei Yu, and Stefan Kaufmann. 2011. Language and ideology in Congress. *British Journal of Political Science* 42(1):31–55.

Dietterich, T. 2000. Ensemble methods in machine learning. *Multiple Classifier Systems* 1–15.

Efron, Bradley, and Gail Gong. 1983. A leisurely look at the bootstrap, the jackknife, and cross-validation. *American Statistician* 37(1):36–48.
Eggers, Andy, and Jens Hainmueller. 2009. MPs for sale? Returns to office in postwar British politics. *American Political Science Review* 103(04):513–33.
Eshbaugh-Soha, Matthew. 2010. The tone of local presidential news coverage. *Political Communication* 27(2):121–40.
Fenno, Richard. 1978. *Home style: House members in their districts*. Boston, MA: Addison Wesley.
Frey, Brendan, and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science* 315(5814):972–6.
Gelpi, C., and P. D. Feaver. 2002. Speak softly and carry a big stick? Veterans in the political elite and the American use of force. *American Political Science Review* 96(4):779–94.
Gerber, Elisabeth, and Jeff Lewis. 2004. Beyond the median: Voter preferences, district heterogeneity, and political representation. *Journal of Political Economy* 112(6):1364–83.
Greene, William. 2007. *Econometric analysis*. 6th ed. Upper Saddle River, NJ: Prentice Hall.
Grimmer, Justin. 2010. A Bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases. *Political Analysis* 18(1):1–35.
———. Forthcoming 2012. Appropriators not position takers: The distorting effects of electoral incentives on congressional representation. *American Journal of Political Science*.
Grimmer, Justin, and Gary King. 2011. General purpose computer-assisted clustering and conceptualization. *Proceedings of the National Academy of Sciences* 108(7):2643–50.
Hand, David J. 2006. Classifier technology and the illusion of progress. *Statistical Science* 21(1):1–15.
Hart, R. P. 2000. *Diction 5.0: The text analysis program*. Thousand Oaks, CA: Sage-Scolari.
Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2001. *The elements of statistical learning*. New York, NY: Springer.
Hillard, Dustin, Stephen Purpura, and John Wilkerson. 2008. Computer-assisted topic classification for mixed-methods social science research. *Journal of Information Technology & Politics* 4(4):31–46.
Hofmann, Thomas. 1999. Probabilistic latent semantic indexing. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* 50–7.
Hopkins, Daniel, and Gary King. 2010. Extracting systematic social science meaning from text. *American Journal of Political Science* 54(1):229–47.
Hopkins, Daniel, Gary King, Matthew Knowles, and Steven Melendez. 2010. *ReadMe: Software for automated content analysis*. http://gking.harvard.edu/readme.
Jackman, Simon. 2006. Data from Web into R. *The Political Methodologist* 14(2):11–6.
Jain, A. K., M. N. Murty, and P. J. Flynn. 1999. Data clustering: A review. *ACM Computing Surveys* 31(3):264–323.
Jones, Bryan, John Wilkerson, and Frank Baumgartner. 2009. *The policy agendas project*. http://www.policyagendas.org.
Jurafsky, Dan, and James Martin. 2009. *Speech and natural language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River, NJ: Prentice Hall.
Jurka, Timothy P., Loren Collingwood, Amber Boydstun, Emiliano Grossman, and Wouter van Atteveldt. 2012. RTextTools: Automatic text classification via supervised learning. http://cran.r-project.org/web/packages/RTextTools/index.html.
Kellstedt, Paul. 2000. Media framing and the dynamics of racial policy preferences. *American Journal of Political Science* 44(2):245–60.
Krippendorff, Klaus. 2004. *Content analysis: An introduction to its methodology*. New York: Sage.
Krosnick, Jon. 1999. Survey research. *Annual Review of Psychology* 50(1):537–67.
Laver, Michael, and John Garry. 2000. Estimating policy positions from political texts. *American Journal of Political Science* 44(3):619–34.
Laver, Michael, Kenneth Benoit, and John Garry. 2003. Extracting policy positions from political texts using words as data. *American Political Science Review* 97(02):311–31.
Lodhi, H., C. Saunders, J. Shawe-Taylor, N. Christianini, and C. Watkins. 2002. Text classifications using string kernels. *Journal of Machine Learning Research* 2:419–44.
Loughran, Tim, and Bill McDonald. 2011. When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *Journal of Finance* 66(1):35–65.
Lowe, Will. 2008. Understanding wordscores. *Political Analysis* 16(4):356–71.
Lowe, Will, Ken Benoit, Slava Mihaylov, and M. Laver. 2011. Scaling policy preferences from coded political texts. *Legislative Studies Quarterly* 36(1):123–55.
MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1: 281–97. London, UK: Cambridge University Press.
Manning, Christopher, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge, UK: Cambridge University Press.
Maron, M. E., and J. L. Kuhns. 1960. On relevance, probabilistic indexing, and information retrieval. *Journal of the Association for Computing Machinery* 7(3):216–44.
Martin, Lanny, and Georg Vanberg. 2007. A robust transformation procedure for interpreting political text. *Political Analysis* 16(1):93–100.
Mayhew, David. 1974. *Congress: The electoral connection*. New Haven, CT: Yale University Press.
Mikhaylov, S., M. Laver, and K. Benoit. 2010. Coder reliability and misclassification in the human coding of party manifestos. 66th MPSA annual national conference, Palmer House Hilton Hotel and Towers.

Monroe, Burt, and Ko Maeda. 2004. Talk's cheap: Text-based estimation of rhetorical ideal points. Paper presented at the 21st annual summer meeting of the Society of Political Methodology.
Monroe, Burt, Michael Colaresi, and Kevin Quinn. 2008. Fightin' words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis* 16(4):372.
Mosteller, F., and D. L. Wallace. 1963. Inference in an authorship problem. *Journal of the American Statistical Association* 58:275–309.
Neuendorf, K. A. 2002. *The content analysis guidebook*. Thousand Oaks, CA: Sage Publications, Inc.
Ng, Andrew, Michael Jordan, and Yair Weiss. 2001. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems 14: Proceeding of the 2001 conference*, eds. T. Dietterich, S. Becker, and Z. Gharamani, 849–56. Cambridge, MA: The MIT Press.
Pang, B., L. Lee, and S. Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing* 10:79–86.
Pennebaker, James, Martha Francis, and Roger Booth. 2001. *Linguistic inquiry and word count: LIWC 2001*. Mahway, NJ: Erlbaum Publishers.
Poole, Keith, and Howard Rosenthal. 1997. *Congress: A political-economic history of roll call voting*. Oxford, UK: Oxford University Press.
Porter, Martin. 1980. An algorithm for suffix stripping. *Program* 14(3):130–37.
Quinn, Kevin. 2010. How to analyze political attention with minimal assumptions and costs. *American Journal of Political Science* 54(1):209–28.
Schrodt, Philip. 2000. Pattern recognition of international crises using Hidden Markov Models. In *Political complexity: Nonlinear models of politics*, ed. Diana Richards, 296–328. Ann Arbor, MI: University of Michigan Press.
Schrodt, Philip A. 2006. Twenty years of the Kansas event data system project. *Political Methodologist* 14(1):2–6.
Slapin, Jonathan, and Sven-Oliver Proksch. 2008. A scaling model for estimating time-series party positions from texts. *American Journal of Political Science* 52(3):705–22.
Spirling, Arthur. 2012. US treaty-making with American Indians. *American Journal of Political Science* 56(1):84–97.
Spirling, Arthur, and Iain McLean. 2007. UK OC OK? Interpreting optimal classification scores for the UK House of Commons. *Political Analysis* 15(1):85–96.
Stewart, Brandon M., and Yuri M. Zhukov. 2009. Use of force and civil–military relations in Russia: An automated content analysis. *Small Wars & Insurgencies* 20:319–43.
Stone, Phillip, Dexter Dunphy, Marshall Smith, and Daniel Ogilvie. 1966. *The general inquirer: A computer approach to content analysis*. Cambridge, MA: The MIT Press.
Taddy, Matthew A. 2010. Inverse regression for analysis of sentiment in text. *Arxiv preprint arXiv:1012.2098*.
Turney, P., and M. L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)* 21(4):315–46.
van der Laan, Mark, Eric Polley, and Alan Hubbard. 2007. Super learner. *Statistical Applications in Genetics and Molecular Biology* 6(1):1544–6115.
van der Vaart, A. W., S. Dudoit, and M. J. van der Laan. 2006. Oracle inequalities for multifold cross validation. *Statistics and Decisions* 24(3):351–71.
Venables, W. N., and B. D. Ripley. 2002. *Modern applied statistics with S*. 4th ed. New York: Springer.
Wallach, Hanna, Lee Dicker, Shane Jensen, and Katherine Heller. 2010. An alternative prior for nonparametric Bayesian Clustering. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)* 9: 892–99.
Weber, Robert P. 1990. *Basic content analysis*. Newbury Park, CA: Sage University Paper Series on Quantitative Applications in the Social Sciences.
Weingast, Barry, Kenneth Shepsle, and Christopher Johnsen. 1981. The political economy of benefits and costs: A neoclassical approach to distributive politics. *The Journal of Political Economy* 89(4):642.
Yiannakis, Diana Evans. 1982. House members' communication styles: Newsletter and press releases. *The Journal of Politics* 44(4):1049–71.
Young, Lori, and Stuart Soroka. 2011. Affective news: The automated coding of sentiment in political texts. *Political Communication* 29(2):205–31.