Advance Access publication February 4, 2015

Political Analysis (2015) 23:254–277
doi:10.1093/pan/mpu019

Computer-Assisted Text Analysis for Comparative Politics

Christopher Lucas
Department of Government and Institute for Quantitative Social Science, Harvard University, 1737 Cambridge St., Cambridge MA 02138, USA
e-mail: clucas@fas.harvard.edu

Richard A. Nielsen
Department of Political Science, Massachusetts Institute of Technology, 77 Massachusetts Avenue Cambridge, MA 02139, USA
e-mail: rnielsen@mit.edu

Margaret E. Roberts
Department of Political Science, University of California, San Diego, 9500 Gilman Drive, #0521 La Jolla, CA 92093, USA
e-mail: meroberts@ucsd.edu

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

Alex Storer
Graduate School of Business, Stanford University, 655 Knight Way, Stanford, CA 94305, USA
e-mail: astorer@stanford.edu

Dustin Tingley
Department of Government and Institute for Quantitative Social Science, Harvard University, 1737 Cambridge St., Cambridge, MA 02138, USA
e-mail: dtingley@gov.harvard.edu
(corresponding author)

Edited by Betsy Sinclair

Recent advances in research tools for the systematic analysis of textual data are enabling exciting new research throughout the social sciences. For comparative politics, scholars who are often interested in non-English and possibly multilingual textual datasets, these advances may be difficult to access. This article discusses practical issues that arise in the processing, management, translation, and analysis of textual data with a particular focus on how procedures differ across languages. These procedures are combined in two applied examples of automated text analysis using the recently introduced Structural Topic Model. We also show how the model can be used to analyze data that have been translated into a single language via machine translation tools. All the methods we describe here are implemented in open-source software packages available from the authors.

Authors' note: Our thanks to Sam Brotherton and Jetson Leder-Luis for research assistance and Amy Catilinac for discussion about text analyses in comparative politics. We also thank Christopher Blattman, Dan Corstange, Macartan Humphreys, Amaney Jamal, Gary King, Helen Milner, Tamar Mitts, Brendan O'Connor, Arthur Spirling, and the Columbia University Comparative Politics Workshop for comments. Our software discussed in this article is open source and available.

1 Introduction

In this article, we focus on new tools for comparativists to utilize *textual* data that can come in many different languages. Massive amounts of textual data are now available to comparativists, from debates in legislative bodies to newspapers to online social media. But using automated content analysis for comparative politics presents important challenges and opportunities, including processing and analyzing text in multiple languages and incorporating data we have about our texts directly into our analyses.

Comparativists are not unfamiliar with tools for textual analysis. Many of the automated text analysis innovations within political science were developed by comparativists (e.g., Schrodt and Gerner 1994; Laver, Benoit, and Garry 2003; Slapin and Proksch 2008). After briefly orienting the reader to the range of text analysis methods available, we highlight a particular approach, unsupervised topic modeling. For interested readers, an online appendix provides an extensive discussion of supervised, scaling, and unsupervised methods to help readers understand the differences between existing approaches and identify methods that will be helpful for their own projects.

To showcase the potential of topic modeling for comparative politics, we use the Structural Topic Model (STM) (Roberts et al. 2013; Roberts, Stewart, Tingley et al. 2014; Roberts et al. 2015) to analyze Arabic *fatwas* and a novel multilanguage analysis of social media responses in Arabic and Chinese to the Edward Snowden event in June 2013. We argue in this article that the STM should be an important part of the text analysis tool kit for comparativists. The STM provides a flexible way to incorporate "metadata" associated with the text, such as when the text was written, where (e.g., which country) it was written, who wrote it, and characteristics of the author, into the analysis using document-level covariates. In turn, it allows comparativists to understand relationships between metadata and topics in their text corpus.

An additional contribution of this article is to discuss a range of tools that are necessary to analyze text from different languages. This includes a discussion of how text processing can differ across languages, along with discussion of robust software tools that properly account for differences across languages. We also consider how to simultaneously analyze text in different languages. In doing so, we discuss multilingual approaches to text analysis, briefly introduce a new `R` package, `translateR`, to access the Google and Microsoft machine translation APIs, and present a novel way to use the STM in a multilingual setting.

The structure of the article is as follows. Section 2 discusses *research questions* in comparative politics that have benefited from text analysis tools, a multilanguage view of *text processing*, and new tools for machine translation. Section 3 presents a brief review of *text analysis tools* with a particular focus on multilanguage text modeling, and introduces the basics of the STM. Section 4 provides two example analyses using the STM: the first looks at Islamic *fatwas* and the second illustrates a novel way to use the model on machine-translated data, with an application to social media responses in Arabic and Chinese to the Edward Snowden event.

2 Text and Language Basics

2.1 *Research Questions and Data Analysis*

Automated content analysis and comparative politics are well suited for each other. Countries around the world are producing textual data at unprecedented rates. Traditional government statistics are often missing, mismeasured, or manipulated, creating a strong incentive for scholars to turn to other forms of data. Meanwhile, governments in almost all countries produce and store large amounts of text data that can be used for descriptive and causal inference. As internet connectivity rises, documents produced by individual citizens are becoming available from an increasingly diverse set of countries. E-mail and advances in survey technologies allow researchers to more easily collect interviews from politicians and government officials, expanding researchers' collections of qualitative data. The digitization of archives, historical records, and public documents has exposed the inner workings of governments across the globe to the public eye.

While other disciplines are only recently catching on to text as a data source, scholars in comparative politics have been using text as data for years, and have built up intuitions for how text

should be used for scholarly inference. Scholars of comparative politics have drawn information from archives and interviews and therefore know how to ask political questions with these data, select important text or interview questions, and find meaningful patterns within the data (George and Bennett 2005; Brady and Collier 2010).

Scholars in comparative politics have already begun using automated methods for analyzing text to ask important political questions. Perhaps the most readily available form of text on politicians, scholars have been using records of speeches politicians make or deliberations among politicians to better understand the internal political workings of governments. Stewart and Zhukov (2009) use public statements by Russian leaders to understand how military versus political elites influence Russia's decision to intervene in neighboring countries. Baturo and Mikhaylov (2013) use federal and sub-national legislative addresses in Russia to identify leadership patterns within the Russian government. Schonhardt-Bailey (2006) uses a text-clustering method to analyze thousands of pages of parliamentary debates in the United Kingdom to analyze the discussion about the repeal of the Corn Laws in Britain. Eggers and Spirling (2011) use parliamentary debates to model exchanges among politicians in the British House of Commons. Miller (2013) analyzes speeches in the United Nations to show that speeches by delegations from countries that were previously colonized devote more words to themes of victimization than states that were never colonized.

Others have tried to infer the policy positions of political parties or political leaders based on documents describing their positions on policies. The Comparative Manifestos project has collected electoral manifestos from all over the world, allowing scholars to use these text data to answer comparative questions about political systems (Budge 2001). Early versions used human coding, but more recently the Comparative Manifestos project and related projects have been assisted by computer techniques. Catalinac (2014) uses thousands of Japanese election manifestos from 1986 to 2009 to determine how electoral strategies shifted after Japan's electoral reform in 1994. Nielsen (2013) uses *fatwas* from websites of Muslim clerics to measure the level of Jihadist thought in these clerics' writings and understand the drivers of Jihadism.

Political scientists have studied newspapers in various languages to ask questions about media freedom and infer relationships between politicians and groups within a country. Van Atteveldt, Kleinnijenhuis, and Ruigrok (2008) analyze Dutch newspapers and extract relationships among political leaders and groups. Coscia and Rios (2012) use news to measure criminal activity in Mexico. Stockmann (2012) studies Chinese newspapers to study how media marketization influences anti-American sentiment in the Chinese media.

Finally, scholars in comparative politics have used blogs and social media sources. King, Pan, and Roberts (2013) studied the focus of censorship in social media in China; Jamal et al. (n.d.) studied anti-Americanism in Arabic-language Twitter posts, and Barberá (2012) used Twitter posts to scale citizen liberal-conservative ideal points across the United States and several European countries. These papers demonstrate an emerging trove of data being generated around the world. With more and more political discourse happening in these forums, comparative politics will require tools that can handle large volumes of data and systematic frameworks to analyze the data.

2.2 *Text Processing Basics: A Multilanguage View*

In order to use automated methods to analyze text, first the analyst must ensure the text is machine-readable. Statistical methods for text analysis are often language agnostic, but the tools for preprocessing the texts are not. This can be challenging for newcomers in comparative politics, as introductions to text analysis often focus exclusively on methods and software for English texts. We discuss three challenges that must be overcome that are particularly important when working with multiple languages within or across research projects: dealing with encodings, preprocessing for dimensionality reduction, and handling large corpora. Along the way, we point out language-specific variations that comparativists studying particular countries should consider. In order to focus our discussion on less well-known issues that come up when working outside English, we leave to Online Appendix A a more general discussion of topics that are more basic, such as the use of Optical Character Recognition. We discuss how we follow these procedures within our sections, where we give examples.

2.2.1 Dealing with encodings

The encoding of text is the way in which the computer translates individual, unique characters into bytes. Each language can have multiple encodings[1] and different computers and different softwares will default recognize different encodings. If the analyst is pulling data from multiple different sources, such as different webpages, it is likely that the text will be in different encodings. In this case, it is necessary to convert each document so that all of the encodings match.[2] The second step is to make sure the software reads the encoding correctly. This can often be done by changing the preference of the software, or encoding the text so that it matches the software's default encoding.

2.2.2 Preprocessing to extract the most information

Automated text analysis methods usually treat documents as a vector containing the count of each word type within the document, disregarding the order in which the words appear. This "bag-of-words" assumption reduces the dimension of natural language text, representing each document as a single vector with length equal to the number of unique words in the text. Unfortunately, even these dictionaries can be too large to be practical, ranging from thousands to millions of unique words. Fortunately, because most words appear only a few times in the corpus, removing infrequently occurring words can dramatically reduce the number of unique word types while having only a small impact on the number of tokens. Bounding the size of the vocabulary can play an important role in helping methods perform well in practice.

In this section, we describe the most common tools for preprocessing textual data, including stop word removal, stemming, lemmatization, compounding, decompounding, and segmentation. In each case, the goal is to reduce the scale of the problem by treating words with very similar properties identically and removing words that are unnecessary to our interpretation and our model. Along with disregarding word order, the so-called "bag-of-words" assumption,[3] these procedures are common preprocessing steps but can differ across languages.

Stop word removal. To aid in interpretation and model performance, analysts often remove words that are extremely common but unrelated to the quantity of interest. These "stop words" are dropped before the analysis. In most settings, this involves removing frequently occurring function words such as "and" and "the," but often removes other types of stop words such as contractions.[4] Most languages have lists of common "stop words" that can be provided to preprocessing programs we discuss below.

We note that for every language, choosing which stop words should be removed is a substantive decision that in some cases can have important effects on the results of the analysis. For example, Campbell and Pennebaker (2003) studied the importance of pronouns, which could be considered stop words in some schemes. Fokkens et al. (2013) found that differing removal of stop words can produce different results in some cases. In other words, choosing a stop word list should be carefully chosen, based on words that the analyst thinks will not be important in informing the analysis. We discuss how we use stop words in more detail in the specific applications (both multilingual and single language) below.

Stemming and lemmatization. Stemming removes the endings of conjugated verbs or plural nouns, leaving just the "stem," which in many languages is common to all forms of the word. Stemming is useful in any language that changes the end of the word in order to convey a tense or

[1]For example, Chinese has several dozen encodings, the largest of which are Guobiao (GB), which has a 2- or 4-byte encoding, Big5 which has a 1- or 2-byte encoding, and ISO-2022, which has a 7-byte encoding.
[2]Most programming languages have packages to transfer between encodings. For example, to convert encodings, we use Python's package *chardet*.
[3]See Online Appendix A for additional discussion.
[4]In other settings, such as the analysis of style or authorship detection, function words may be the sole quantity of interest (Mosteller and Wallace 1963).

number, which includes English, Spanish, Slovenian, French, modern Greek, and Swedish. Since tense and number are generally not indicative of the topic of the text, combining these terms can be useful for reducing the dimension of the input. However, not all languages require stemming. For example, Chinese verbs are not conjugated and nouns in Chinese are usually not pluralized by adding an ending. A host of studies have shown stemming to be an effective form of preprocessing in English; however, the benefits are both application- and language-specific (Salton 1989; Harman 1991; Krovetz 1995; Hull 1996; Hollink et al. 2004; Manning, Raghavan, and Schutze 2008).[5]

Stemming is an approximation to a more general goal called lemmatization—identifying the base form of a word and grouping these words together. However, instead of chopping off the end of a word, lemmatization is a more complicated algorithm that identifies the origin of the word, only returning the *lemma*, or common form of the word. Lemmatization can also determine the context of the word; for example, it will leave *saw* the noun as is, but will turn *saw* the verb into *see* (Manning, Raghavan, and Schutze 2008). While stemming often works almost as well as lemmatization in languages like English, lemmatization works better for languages where conjugations are not indicated by changing the end of the word, and for agglutinative languages[6] where there is a greater variety of forms for each individual word, such as Korean, Turkish, and Hungarian.

Compound words. Some languages will frequently concatenate two words that describe two different concepts, or split one word that describes one concept. These instances, called compound words or decompounded words, can decrease the efficacy of text analysis techniques because one concept can be hidden in many unique words, or one concept may be split across two words. For example, the German word "Kirche," or church, can be appended to "rat," forming "Kirchenrat," who is a member of the church council, or "pfleger" to form "Kirchenpfleger," or church warden. If it is appended, the computer will not see "Kirch" as an individual concept. Decompounding this case would separate "Kirch" from its endings. "Compounding languages" include German, Finnish, Danish, Dutch, Norwegian, Swedish, and Greek (Alfonseca, Bilac, and Pharies 2008). On the other hand, the analyst may want to compound words. For example, in English, "national security" and "social security" each contain two separate terms even though they express one concept. Even though they share the word "security," these concepts are very different from each other, so the analyst might wish to compound these into "nationalsecurity" and "socialsecurity." All of these decisions should be guided by substantive knowledge.

Segmentation. Some languages, like Chinese, Japanese, and Lao, do not have spaces between words and therefore text analysis techniques that rely on the word as the unit of analysis cannot naturally parse the words into individual units. Automatic segmentation must be used before the documents can be processed by a statistical program (see Lunde [2009] for an overview). Segmentation can be done using dictionary methods (Cheng, Young, and Wong 1999) or using statistical methods that learn where spaces are likely to occur between words (Tseng et al. 2005).

2.2.3 Building the document-term matrix

Once all preprocessing has been completed, for many automated content techniques (including those detailed in this article), the remaining words are used to construct a document-term matrix (DTM). A DTM is a matrix where each row represents a document and each column represents a unique word. Each cell in the matrix denotes the number of times the word indicated by the column appears in the document indicated by the row. For example, if a document was just the sentence "I support the Tories," "I" and "the" would likely have already been removed as stop words, so that the document would be represented with a 1 for "Tories" and "support" and a 0 for all other words.

[5]Several computer programs are available to implement stemming, including `txtorg` (discussed in Section 2.3), which can implement stemming in multiple different languages. These programs automatically detect common variations in word endings, removing these endings, and plural words into their singular form.

[6]Languages where most words are formed by combining smaller meaningful language units called morphemes.

Following the "bag-of-words" assumption, the DTM format preserves information about how many times each word appears in a document while discarding information about the word order. The resulting matrix is extremely sparse, meaning a large proportion of the cells are zeros, because most documents will contain only a small fraction of the words in the vocabulary. For even moderately sized corpora, this matrix will be too large to store in its rectangular form; however, we can exploit the sparsity of the DTM to store only the non-zero entries. The DTM, or its sparse representation, is the primary input to most automated text analysis methods, including the ones in this article.

2.3 *Multilanguage Preprocessing Tools*

2.3.1 Language-specific processing

All of the previous steps are not trivial from a workflow perspective, especially for comparativists working in a variety of languages, each of which may require specialized tools. Here we discuss existing methods to deal with preprocessing text within a language. There are two flexible open-source software tools for doing stemming, stop word removal, etc., that cover many languages. First is the R package `tm` (Feinerer, Hornik, and Meyer 2008), which can stem 11 languages[7] and can do stop word removal on 13 languages.[8] Another tool is the Python/Lucene-based application `txtorg`,[9] which currently includes support for 32 languages.[10] In `txtorg`, all supported languages go through a suite of best practice preprocessing steps, which includes the appropriate combination of stemming, segmentation, and stop word removal for that particular language. Both of these tools facilitate text preprocessing, though `txtorg` is dramatically more efficient in handling larger corpora and when searching and subsetting large amounts of text.[11]

2.3.2 Translation

As we discuss in Section 3.2 and illustrate in Section 4.2, there are important instances where modeling textual data from multilingual corpora becomes more efficient and accessible for applied users if the text is first translated into a single language. Of course, though human translation remains the gold standard, the scale of textual data generally far exceeds that which might be feasibly translated by humans. In subsequent sections, we discuss the relevant technical considerations of multilingual analysis in greater detail. In this section, we briefly discuss machine translation and introduce an `R`-based utility for accessing machine translation software developed by Google and Microsoft.

Central to comparative politics is, of course, a commitment to cross-national comparison. And while comparativists have developed many techniques for automated text analysis, there presently exists little or no support for cross-lingual comparison. While this limitation does not preclude all potentially interesting comparisons, it prevents a great many. In Section 3.3, we discuss a principled way by which such comparisons can be made with the STM after first translating the corpus into a common language. However, this requires first overcoming the potentially formidable task of translating the data into a common language.

The job of a translator is to "render in one language the meaning expressed by a passage of text in another language" (Brown et al. 1990, p. 81), and though there exist many approaches, the basic task of machine translation is to accomplish this conversion with a computer. Because of its many uses and because early barriers to machine translation, which included hardware limitations and a

[7]Danish, Dutch, English, Finnish, French, German, Norwegian, Portuguese, Russian, Spanish, and Swedish.
[8]Danish, Dutch, English, Finnish, French, German, Hungarian, Italian, Norwegian, Portuguese, Russian, Spanish, and Swedish.
[9]Note that `txtorg` includes a graphical user interface built with `TkInter`, so users do not need to know Python in order to use `txtorg`. Nearly all `txtorg` functionality is accessible without writing any code.
[10]Arabic, Armenian, Basque, Bulgarian, Catalan, Chinese, Czech, Danish, Dutch, English, Finnish, French, Galician, German, Greek, Hindi, Hungarian, Indonesian, Irish, Italian, Japanese, Korean, Latvian, Norwegian, Persian, Portuguese (separate tools for Brazil and Portugal), Romanian, Russian, Spanish, Swedish, Thai, and Turkish.
[11]See Online Appendix G for some basic benchmarking information between `tm` and `txtorg`.

dearth of machine-readable text (Brown et al. 1993), have largely been overcome, there is now heavy investment in machine translation. There exist a number of academic and commercial labs committed to the development of machine translation systems, some of which have led to the founding of new companies. Simultaneously, large, mature software companies like IBM, Microsoft, and Google have also developed their own machine translation systems (Koehn 2009). Because these groups can leverage financial and academic resources beyond those generally accessible to political scientists, we argue that a desirable solution to the problem of machine-translating text for political science is one that leverages the effort made by dedicated research groups in a simple, straightforward way.

When translating text for eventual consumption by human readers, there can be no substitute for human translation. Within the literature on machine translation evaluation, it is said that "The closer a machine translation is to a professional human translation, the better it is" (Papineni et al. 2002, p. 1). But compared to translating text for eventual consumption by human readers, translation for multilingual text analysis is a slightly easier problem. As discussed in Section 2.2, most approaches to automated text analysis make a bag-of-words assumption, which implies that the ordering of terms in a document does not matter. The translation software needs only to correctly translate the significant terms in the original document, as any error in word order will be discarded by the bag-of-words assumption.

If users want to use machine translation, what should they use? Our answer is to provide an `R` package, `translateR`, that permits easy access to two very mature translation systems, namely those produced by Google and Microsoft. The package supports a variety of input and output formats and can be easily used with other text analysis software. Crucially, for our purposes, the package preserves information about individual texts (such as the original language or date of authorship). This is important for using models like the STM that incorporate these data. Moreover, `translateR` preserves the scalability of machine translation by the translation process via multiple API calls. Users provide as input the data to be translated, either as a dataframe with metadata or as a vector of documents or terms, and `translateR` makes calls in parallel to the translation API specified by the user (either Bing or Google). As a result, researchers spend minimal time reformatting their data and similarly little time waiting for the translation process to finish along with other aspects necessary for standard textual analysis. Additional discussion and syntax are given in Online Appendix C.

3 Computer-Assisted Text Analysis

In the previous section, we discussed in detail how to prepare a multilingual corpus for automated approaches to text analysis by creating a DTM. A complete overview of methods for quantitatively analyzing the text is beyond the scope of this article. Unlike the issues involved in multilingual text processing, these methods have been well developed elsewhere (e.g., Grimmer and Stewart 2013). In Section 3.1, we provide a brief, selective overview and direct interested readers to our online appendix, which provides an accessible introduction to a broader range of methods. We then discuss the challenges that arise in moving from single to multilingual corpora (Section 3.2). Finally, in Section 3.3, we describe the STM before providing two applications of its use (Section 4).

3.1 *A Brief Overview of Approaches*

There are essentially two approaches to automated text analysis: supervised and unsupervised methods, each of which *amplifies* human effort in a different way. In *supervised* methods, we specify what is conceptually interesting about documents in advance, and then the model seeks to extend our insights to a larger population of unseen documents. Thus, for example, we might manually classify 100 documents into two categories, with the model classifying the remaining 9900 documents in the corpus. In *unsupervised* methods, such as topic modeling, we do not specify the conceptual structure of the texts beforehand. Instead, we use the model to find a low-dimensional summary that best explains observed documents given some set of assumptions. Consequently,

human effort shifts from construction of a training set in supervised learning to interpretation of the model results in unsupervised settings.

In our applications, we leverage a particular type of unsupervised topic modeling built on the popular Latent Dirichlet Allocation (LDA) model (Blei 2012). LDA is a mixed-membership model, which means that each document is represented as a mixture over a set of topics and each observed word is conditionally independent given its topic.[12] Each topic is a distribution over the words in the vocabulary, which crucially are learned rather than assumed by the model. LDA has seen widespread use in computer science and the humanities due to its simple and extensible structure.

The full range of text analysis methods including supervised and unsupervised methods is discussed in greater detail in Grimmer and Stewart (2013). We have also included an online appendix for this article containing an abbreviated introduction using a consistent set of heuristic examples using data from a corpus of comparative politics papers published in the *American Political Science Review* (Online Appendix B).

3.2 *Multilingual Text Modeling*

A considerable advantage to the quantitative approach to text analysis is that the methods are language agnostic. However, a rarely discussed limitation is that the documents are assumed to be drawn from only one language. This can be a frustrating situation for practitioners in comparative politics who are interested in studying a multilingual corpus. Here we discuss the attendant methodological issues that apply to both supervised and unsupervised models.

In some respects, the most natural approach for handling a multilingual corpus is to perform analysis within the native language but referencing a commonly shared objective. This is the approach taken in manual coding efforts, such as the Comparative Manifestos Project (Volkens et al. 2013), where it is relatively straightforward to define the coding criteria in a language-independent way but analyze each document in its own native language. For keyword and supervised approaches, it is plausible to develop a separate but statistically comparable dictionary or training set for each observed language. Unlike the manual case where a single codebook can be developed in a shared language, the automated approaches require a duplication of effort for each language. While feasible in supervised settings, there is not a clear analogue for unsupervised methods.

A second approach is to translate text into a common language. Manual translation by an experienced translator would be extremely costly and so we turn to machine translation tools introduced above. How well this works will depend on the quality of the machine translation and the goal of the analysis. We return to this approach in Section 4.2.

The third approach is to develop a model which maintains an explicitly multilingual representation. The central challenge is to develop an alignment between the conceptual representations of the model across languages so that we know a particular scaling, topic, or class in one language is comparable with the representation in another language. We focus here on the challenging case of unsupervised topic models in the style of LDA, where the conceptual representation is being learned from the data, although the general ideas apply straightforwardly to supervised methods as well.

Existing approaches to multilingual topic models are differentiated in how they leverage external information to implicitly or explicitly align comparable topics across languages. The Polylingual Topic Model of Mimno et al. (2009) leverages a set of aligned documents, for example Wikipedia articles on the same topic in different languages. By constraining aligned documents to share a distribution over topics, the model is able to align the words associated with a given topic across languages. The Bilingual Topical Admixture model (Zhao and Xing 2006) works with texts which are aligned at the token level (such as through the result of machine translation). Exact translations which are aligned at the token level are more difficult to obtain, but they provide a more direct source of information about topic alignment. Finally, the Multilingual Supervised LDA model (Boyd-Graber and Resnik 2010) uses a combination of sentiment information and aligned dictionaries to develop multilingual topics. Recent work combines these approaches to leverage

[12]By "mixture" in this context we mean a set of positive values that sum to one.

both dictionary- and document-level alignments simultaneously, resulting in a model which is more robust than either independently (Hu et al. 2014).[13]

From a technical perspective, fitting most of these models involves a relatively straightforward adaptation of the collapsed Gibbs sampling algorithm for LDA (Griffiths and Steyvers 2004).[14] The result is a set of topics for each language along with the document-topic loadings. Multilingual models have primarily been used for either document exploration or machine-translation tasks.

The existing models for multilingual analysis do, of course, have limitations. The correspondence between the multilingual topics relies on the particular alignment information provided by the user and needs to be validated. This can be particularly challenging for indirect strategies such as the document alignment in the Polylingual Topic Model. For each topic, the user needs to verify that the topic word distributions are comparable across languages. Given that the size of the vocabulary may be in the thousands, assessing model failure can be a substantial challenge even for only two languages.[15] While the articles described above provide diagnostic tools for the model results, they are primarily focused on the machine-translation applications that motivate that literature.

As a practical matter, multilingual topic models generally lack the ability to include additional document metadata, which we argue below is an important part of applied social science research. In addition, there are limited software tools available for the estimation of these models.[16] These critiques are not problematic for the models as presented in their original context, but do suggest challenges for their use in applied comparative research. Below, we suggest a way that machine translations and the STM can be fruitfully combined.

3.3 *The STM*

In our applications (Section 4), we leverage a recently introduced framework, the STM (Roberts et al. 2013; Roberts, Stewart, Tingley et al. 2014; Roberts, Stewart, and Airoldi n.d.) The STM is a mixed-membership topic model (like LDA) with extensions that facilitate the inclusion of document-level metadata.[17] The inclusion of this information within the model can both improve the quality of the learned topics and facilitate hypothesis testing. Software for estimating the model is freely available in the `R` package `stm`.

Before moving on to our applications of STM, we first briefly review several aspects of our use of the STM which are specific to this context. A brief statement of the model is available in Online Appendix D. For additional technical details on estimation and implementation of the model, we refer to existing work (e.g., Roberts et al. 2013; Roberts, Stewart, and Tingley et al 2014; Roberts, Stewart, and Airoldi n.d.).

The role of covariates. STM differs from other topic-modeling techniques like LDA in allowing document-level covariates to be included in the model as a method for pooling information. A covariate can be allowed to affect either *topical prevalence* or *topical content*. Covariates on

[13]Boyd-Graber and Blei (2009) introduce a topic model for completely unaligned texts, but they note that the model is highly sensitive to starting values and when run to divergence can result in the nominally equivalent topics between languages diverging. This is evidence for the central role of observed alignment information in pinning down the correspondence between topics.

[14]For example, for the Polylingual Topic Model, we iteratively sample each token in the document, adjusting the topic-word distribution for the language-specific version of the topic but sharing the document-topic counts across all languages within the document. This algorithm has comparable speed to LDA but with slightly higher memory requirements.

[15]As the number of languages grows, this problem is compounded by the need to have a single scholar who reads all languages. For example, among our team, no author speaks both Arabic and Chinese, which would make direct validation of a Polylingual Topic Model quite difficult.

[16]Of the models discussed here, only the Polylingual Topic Model of Mimno et al. (2009) has a publicly available software implementation. A Java implementation is available in the software package Mallet (McCallum 2002).

[17]The inclusion of document metadata follows and extends two developments within political science. The Dynamic Topic Model (Quinn et al. 2010) is a single-membership model in which the probability of observing a topic moves smoothly through time. The Expressed Agenda Model (Grimmer 2010) is a single-membership model which includes information about document authors. However, no such model exists to include author and time simultaneously. Drawing on these works, our approach generalizes to arbitrary covariate information and extends these setups for the mixed-membership case.

topical prevalence allow documents to share information about which topics are expressed within the document (e.g., women are more likely to talk about topic 1 than men). Users can plot the relationship between their topic prevalence covariates and the expected proportion of a document that belongs to each topic. Covariates on topical content allow for the rates of word use, for each topic, to differ by covariate values (e.g., women are more likely to use a particular word when talking about a particular topic than men). Users can include both prevalence and content covariates, only one type, or neither.

Content covariates are a particularly powerful tool which can be used to capture both quantities of interest and condition away systematic differences within the corpus that are not of primary interest. Imagine, for example, we were attempting to compare topical coverage within a large corpus of news reports about China from Agence France Presse (AFP) and Xinhua, China's state news agency. In order to facilitate a direct comparison, we want the model to discover (for example) a single topic on Tibet; however, systematic differences in the way that AFP and Xinhua cover Tibet may produce separate AFP-Tibet and Xinhua-Tibet topics. Instead, by allowing the model to maintain an AFP version of the topic and a Xinhua version of the topic (which are constrained to be close), we can estimate the differences in word use and still retain a straightforward comparison. If the differences are themselves of interest, the analyst can compare words distinctive to the Xinhua version of the topic ("oil," "gas," "resources") with words distinctive of the AFP version ("culture," "religion," "independence").[18] If the differences are simply a nuisance, we can marginalize over source-specific versions of the topics weighting by the document frequency within the corpus as a whole. We will return to this idea in our multilingual analysis, where we use content covariates to condition out systematic differences that result from translation to English from different languages.

Topic correlations. In addition to the inclusion of covariates, the second distinctive feature of the STM is the explicit estimation of correlation between topics.[19] Graphical depictions of the correlation between topics provide insight into the organizational structure at the corpus level. In essence, the model identifies when two topics are likely to co-occur within a document (here we focus on positive correlations although negative correlations are also estimated). The software we provide allows the user to produce a network graph of topics where each topic is a node and two nodes are connected when they are highly likely to co-occur. This can help the user identify larger themes that transcend topics.

Drawing on recent literature in undirected graphical model estimation, we extend the approach developed in Blei and Lafferty (2007) for estimating the edges of the graph. In Online Appendix E, we describe the two graph estimation procedures we provide along with parameters set by the user. We give a specific example of this approach in the next section.

4 Applications

In this section, we introduce two applications of the STM. The first application, the analysis of Islamic *fatwas*, is conducted entirely within the single native language. For the second application, our corpus includes both Chinese and Arabic texts, which we translate into a common language prior to analysis. All the analysis tools used below are built into the R package `stm` (Roberts, Stewart, and Tingley 2014).

4.1 *Jihadi Fatwas*

In this example, we combine data on Muslim clerics from Nielsen (2013) with expert coding of whether clerics are Jihadist or not to see how the topical content of contemporary Jihadist religious

[18]The example here is drawn from the data and model described in Roberts et al. (n.d.).
[19]Correlations are estimated by replacing the Dirichlet distribution in the standard LDA framework with a logistic normal distribution as in the Correlated Topic Model (Blei and Lafferty 2007). When no covariates are specified, the STM reduces to an instance of the Correlated Topic Model.

texts differ from those of non-Jihadists. Nielsen collects data on the lives and writings of 101 prominent Jihadist and non-Jihadist Muslim clerics, including the 27,248 texts available from these authors from online sources. A majority of these texts are *fatwas*—Islamic legal rulings on virtually any aspect of human behavior, ranging from sex and dietary restrictions to violent Jihad. For many clerics, Nielsen also collects books, articles, and sermons on the same types of topics. Collectively, these texts are representative of how clerics choose to interact with religious constituencies; in fact, many of these collections are curated by the clerics themselves.

We combine these texts with an independent coding of whether these clerics are Jihadist or not based on two scholarly sources. First, the *Militant Ideology Atlas—Executive Report* (McCants 2006), Appendix 2, lists 56 individuals that are frequently cited by Jihadists. The authors of the Atlas code whether these are "Jihadi authors" according to substantive knowledge. Second, Jarret Brachman (2009, pp. 26–41) lists the names of prominent clerics in eight ideological categories: establishment Salafists, Madkhali Salafists, Albani Salafists, scientific Salafists, Salafist Ikhwan, Sururis, Qutubis, and Global Jihadists. The latter two categories are Jihadist, whereas the rest are not. These two sources largely overlap; together, they provide expert assessments of 33 of the clerics (20 Jihadists and 13 non-Jihadists) for whom Nielsen collects 11,045 texts.

We then estimate an STM with the binary indicator for Jihadi status as a predictor. The results are shown in Fig. 1, with topics presented as collections of words (in this figure, we leave the words in Arabic), along with the topic coefficients and standard errors. We estimate 15 topics after experimenting with 5- and 10-topic models that produced less readily interpretable topics.[20]

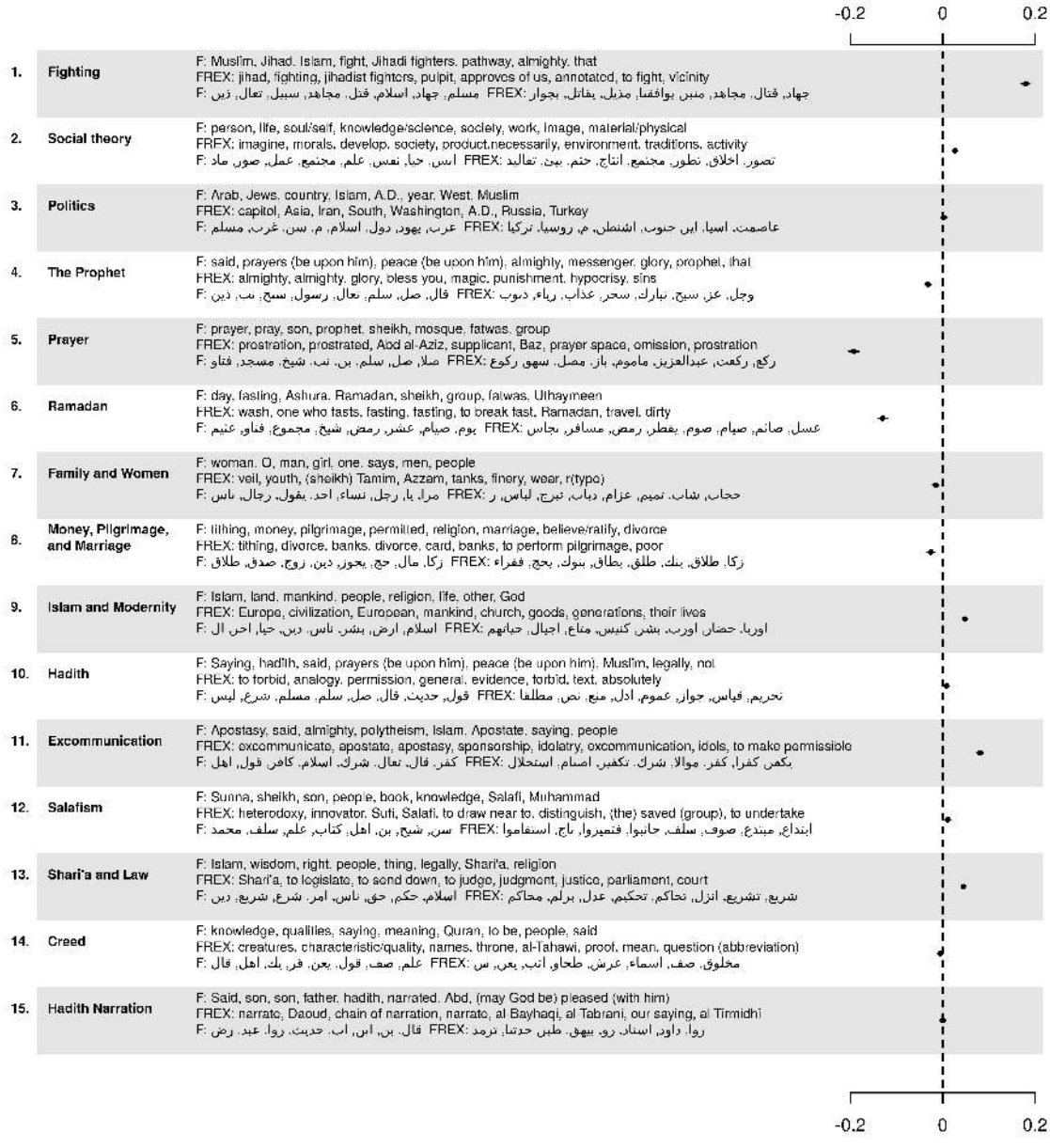
The first inferential task is to infer topic labels from the words that are most representative of each topic. We do this by examining the most frequently occurring words in each topic and the words that have the highest levels of joint frequency and exclusivity (meaning they are common in one topic and rare in others). In several cases, we also examine *exemplar documents* for a topic—those documents that have the highest proportion of words drawn from the topic. This also serves as a validation step because we check whether words in the topic have the meanings in context that they appear to have in the topic frequency lists.

The results in Fig. 1 indicate that topics 1 (Fighting) and 11 (Excommunication) are most correlated with the indicator for Jihadist clerics, matching our *a priori* predictions based on the content of the topics. Excommunication (*takfīr* in Arabic) is commonly used by Jihadists to condemn fellow Muslims who disagree with Jihadist aims or tactics. The exemplar documents for this topic are *fatwas* on the rules and justifications for excommunication and other writings that make heavy use of the concept of excommunication. In contrast, topic 1 is a broader Jihadist topic focused primarily on fighting the West—the exemplar documents are *fatwas* about fighting abroad. Topics on social theory, Islam and modernity, and Shari'a and law are also correlated with Jihadism, though to a lesser degree.

A number of other topics are also clearly identifiable, including topic 5 on prayer, topic 6 on Ramadan, and topic 8 on money, pilgrimage, and marriage. As we expected from their content, these topics receive relatively little attention from Jihadists, who are more focused on their violent struggle than with fine distinctions in Islamic legal doctrine and religious ritual.

We can use the estimated correlation of topics with other topics to learn more about the structure of the corpus.[21] In Fig. 2, we plot the network of topics such that topics that are correlated are linked. Many of the correlations between topics are intuitive and revealing about the nature of Islamic legal discourse. The topic on *hadith* (the sayings of the Prophet Muhammad) is highly correlated with language about the chain narration by which each *hadith* is verified as trustworthy. Authors who write about social theory are likely to also write about Islam and modernity, politics, the role of women, and Shari'a and law.

[20]This is not to say that 15 is the "right" number of topics in this corpus—rather, we find a 15-topic model for uncovering useful insights about the structure of the texts in relation to the Jihadist ideology of their authors.
[21]We introduce our approach to calculating and graphically representing the correlation structure in Online Appendix E.



Note: The x-axis was incorrect in the published version, ranging from -1.25 to 1.25. The correct x-axis is shown here, ranging from -0.2 to 0.2.

Fig. 1 Coefficients and standard errors for a 15-topic Structural Topic Model with Jihadi/not-Jihadi as the predictor of topics in Arab Muslim cleric writings. The words used to label each topic are shown on the left. "F:" indicates words that occur most frequently in each topic. "FREX:" indicates words that are frequent *and* exclusive to each topic. The Arabic words are in their stemmed form.

Figure 2 shows correlations between topics preferred by Jihadists. Documents that include language about excommunication tend to also include text about creed (what Muslims believe), shari'a and law, the Prophet, and fighting. Documents about fighting are likely to also include politics, discussions of Salafism, Islam and modernity, and Shari'a and law. In contrast, texts about non-Jihadi legal issues—prayer, Ramadan, money, pilgrimage, and marriage—are unlikely to be about more than one topic. This aligns with our qualitative assessment of the corpus: the modal Jihadist *fatwa* is article-length and ranges across multiple topics, whereas the modal non-Jihadist *fatwa* is paragraph-length and gives a precise ruling on only one topic.

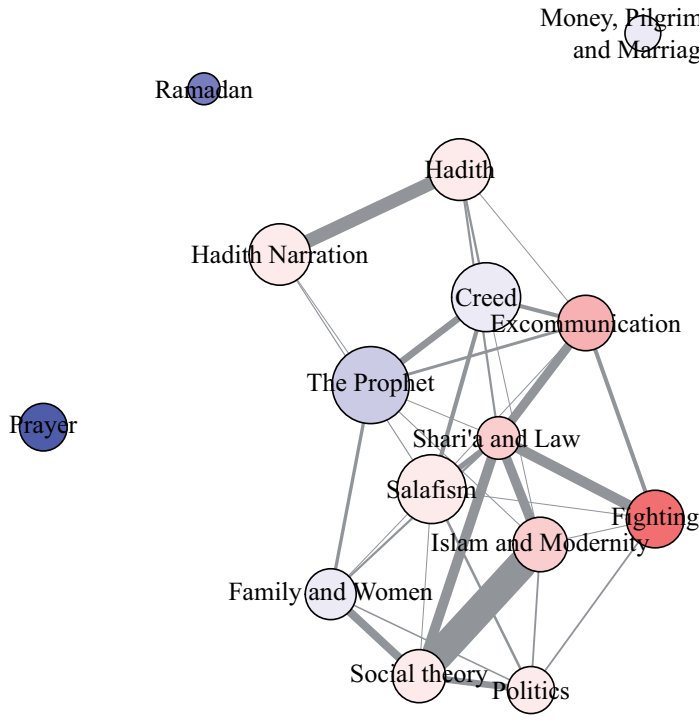


Fig. 2 The network of correlated topics for a 15-topic Structural Topic Model with Jihadi/not-Jihadi as the predictor of topics in Arab Muslim cleric writings. Node size is proportional to the number of words in the corpus devoted to each topic. Node color indicates the magnitude of the coefficient, with redder nodes having more positive coefficients for the Jihadi indicator and blue nodes having more negative coefficients. Edge width is proportional to the strength of the correlation between topics.

The presence of at least two clearly Jihadist topics invites further inquiry. Figure 2 shows that these topics are correlated in general, but do all Jihadists write on both topics? Do some write more on one? Does this split indicate an intellectual divide within the Jihadist subgroup? To take a first cut at these questions, we simply plot the proportion of the *Excommunication* topic against the *Fighting* topic, as shown in Fig. 3. The results teach us several new things about how Jihadists and non-Jihadists write. First, for many Jihadists, document space spent on *Fighting* is substitute for space spent on *Excommunication*.[22] Usama bin Laden has the highest proportion of words devoted to *Fighting*—about 38%—but he spends only 2% of his words discussing the excommunication topic. This accords with Bin Laden's long-time focus on the goal of targeting and provoking the West through both writings and deed.

At the other extreme, Ahmad al-Khalidi and Ali Khudayr, respectively, devote 46% and 32% of their writing to excommunication and almost none to fighting. This is not surprising when we consider the life trajectories of these clerics. Both have issued *fatwas* excommunicating prominent Muslims for alleged heresies and both have spent time in Saudi prisons for doing so. This finding adds further face validity to our findings—the clerics most interested in writing about excommunication of fellow Muslims are those that have also carried it out repeatedly. Between these endpoints, most other Jihadists spread out on a continuum where more discussion of excommunication means less of fighting and vice versa. It is likely that these two topics are virtually all that some of these authors write about. Given that filler words and others must still be assigned to topics, it may simply be the case that no more than 50% of a document can be allocated across these Jihadi topics.

[22]This is not inconsistent with the finding that these two topics are correlated within texts. The presence of one topic increases the likelihood of the presence of the other topic in a text, but some authors focus on one topic more than the other.

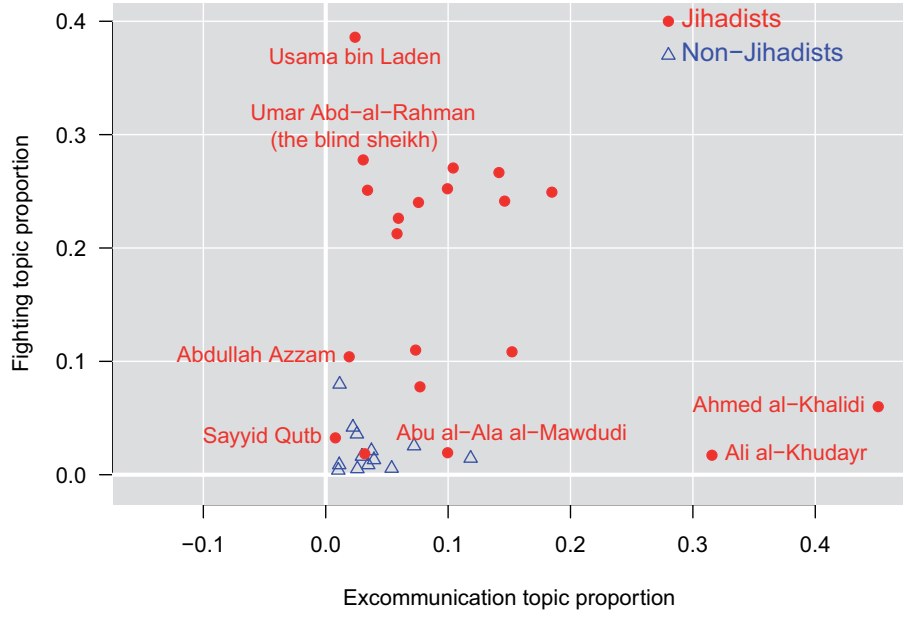


Fig. 3 Estimated topic proportions by fighting the West and excommunication topics, separated out by Jihadist versus Jihadist coding.

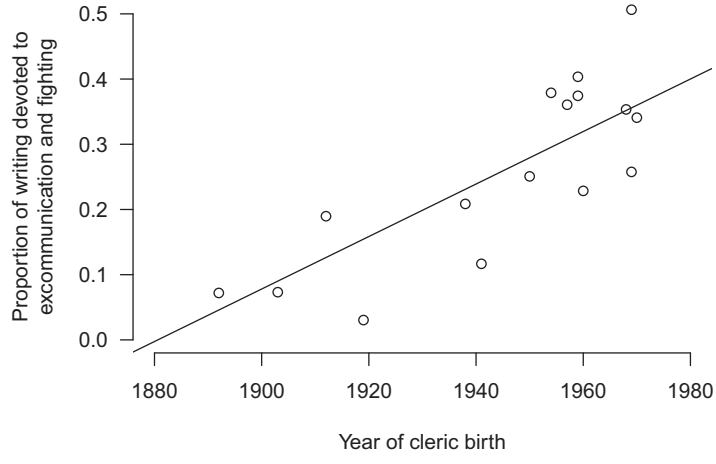


Fig. 4 The proportion of words by each Jihadi author devoted to excommunication or fighting, plotted against the year of their birth with a best-fit line.

Several Jihadist authors have low enough proportions of both Jihadist topics that they could be mistaken for non-Jihadist clerics. Sayyid Qutb is often considered one of the founders of the modern worldwide Jihadist movement, but only 3% of his writing is devoted to the topics that tend to occupy other Jihadists. Similarly, Abu al-Ala' al-Mawdudi and Abdullah Azzam are considered canonical authors by Jihadists, but only about 10% of their writings are devoted to the topics of *fighting* and *excommunication*.

To see what is unique about the writing of these authors, we look at the topics to which they devote the most attention and find that their profiles are very similar. Each devotes the bulk of their writing to writing about social theory, politics, and Islam and modernity. We find that the current Jihadist focus on fighting the West and excommunication is relatively new. We show this in Fig. 4 by summing the proportion of writing that each Jihadist author devotes to either excommunication

or fighting and plotting it against the year of each cleric's birth. Among the set of individuals identified in the secondary literature as Jihadists, only those of relatively recent vintage are writing on the two topics that are now core to Jihadist ideology.

To summarize, we find that a 15-topic model provides insight into the structure of an Islamic legal corpus that includes work by Jihadists and non-Jihadists. Although one might expect Jihadism to be monolithic, there are in fact multiple ways that Jihadists write about their subject. In particular, there is suggestive evidence of a trade-off for many Jihadists between focusing on fighting the West and focusing on excommunicating fellow Muslims they feel are inadequately supporting the Jihadist cause. We also find that an older generation of Jihadist writers does not write about either of these topics, suggesting that Jihadist writing was more eclectic in the past but has become homogenized over time.

4.2 *Reactions to Snowden in China and the Middle East*

In this section, we provide an illustrative example of how machine translation can be used in conjunction with the STM to make comparisons across countries and languages. An important theoretical and empirical agenda is understanding how other countries view the United States (Katzenstein and Keohane 2007; Chiozza 2009; Lynch 2007; Telhami 2002; Rubin 2002). One way to understand views of the United States is to compare responses to specific events (e.g., Jamal et al. n.d.). Here we look at responses to a single event across different language communities. We collected thousands of social media posts in Arabic and Chinese during June 2013, the month when former U.S. government employee Edward Snowden disclosed thousands of classified documents that detailed the U.S. government's clandestine surveillance program. Because the documents leaked by Snowden contained many revelations about surveillance of and cooperation with other countries, some scholars worried that the leaks would undermine U.S. legitimacy abroad (Farrell and Finnemore 2013). We focus on the reaction of citizens in China and the Middle East, arguably two of the most important U.S. strategic areas in the world.

We generate our corpus by collecting the universe of unique posts from Twitter in Arabic containing the word for "Snowden" and the universe of unique posts from Sina Weibo containing the word for "Snowden" in Chinese from June 1 to June 30, 2013.[23] Twitter is banned in China, so a collection of Twitter posts in Chinese would contain those of foreign Chinese speakers, or of those who are sophisticated enough to jump the Great Firewall, and therefore would be a potentially biased sample. Sina Weibo is the closest comparable platform to Twitter in China.[24]

4.2.1 Two approaches to machine translation

Ideally, we want to analyze both Arabic and Chinese within the same topic model. Leaving the two corpora in their respective languages would lead to essentially no overlap in vocabulary between the Arabic and Chinese posts. As a result, each corpus would have its own individual topics, since the model cannot recognize that Snowden in Arabic is the same word as Snowden in Chinese, rendering direct comparison of topical content essentially impossible. As described in Section 3.2, we need to use some type of external alignment between languages to analyze the two corpora within the same model. Translation provides alignment by creating overlap between the two corpora. Here we explore solutions based on machine translation as software implementations are widely available and continuously improving. We use two approaches to machine translation: translating the entire corpus and translating only terms that appear in the DTM. Both approaches easily extend to document sets containing more than two languages.

In the first approach, we use machine translation to translate both corpora of text completely into a common language, English. There are compelling reasons to translate to a "third-party" language, particularly when that language is English. Perhaps the most basic reason for choosing

[23]We point readers interested in the preprocessing that we conducted on the Snowden corpus to Online Appendix F.3.
[24]Both Weibo and Twitter restrict the number of words within posts. All data were obtained from Crimson Hexagon.

English is the ability to communicate research findings to an English-speaking audience. We also wanted both sets of text to undergo the same amount of translation. If we had translated the Arabic into Chinese, for example, and left the Chinese text untranslated, the Chinese corpus might dominate the topic model as it would have no words that were "untranslatable." This at least makes it more plausible that the inevitable error introduced in translation is roughly comparable between the two language groups, resulting in a type of symmetry. Of course, this may not be the case when two languages are more closely related or where the translation accuracy is substantially higher for one kind of transformation.

Beyond the appeal of symmetry, English is a particularly useful common language due to its role in machine-translation systems. Most modern machine translation systems use parallel corpora to learn the parameters of a statistical model. However, many language pairs do not have large parallel corpora easily available and so instead a "pivot," or bridge, language is used as an intermediate point in creating the translation. English is a common pivot language due to the widespread availability of texts. Thus, not only would we expect the Chinese to English and Arabic to English to have particularly high accuracy, but for many machine translation systems a translation between Chinese and Arabic will involve a translation through English.[25] For more on pivot languages in statistical machine translation systems, we refer readers to Utiyama and Isahara (2007) and Paul et al. (2009). Habash and Hu (2009) discuss the specific case of using English as a pivot language for Chinese and Arabic.[26] We use Google Translate to perform translation, passing each post through `translateR` and recording the translation. Online Appendix F discusses our preprocessing steps.

The complete corpus strategy is ideal because it introduces no additional sources of information loss beyond the machine-translation process. Because each original text is translated, words are always considered within the context that they appear. Context not only improves accuracy in most machine-translation systems, but may, in some cases, be necessary for an appropriate translation. The downside is that the process of machine-translating a corpus of even a few thousand documents can be expensive and time-consuming because all the text is passed to the machine-translation service.

Given these considerations, we also investigate a second approach which relies on only the minimal number of translation queries. We first created a DTM for each language's corpus separately and translate only those terms. We take the intersection of the two translated vocabularies and merge the document-term matrices together. While this approach discards word context within translation, it is considerably cheaper.[27] The cost of translation for the complete corpus grows linearly with the size of the corpus because every occurrence of every unique term is translated. By contrast, in the term-by-term translation, the marginal cost of translating an additional document decreases as the corpus grows, because there are fewer and fewer unique terms in each additional document as more documents are added to the corpus.

[25] As a proprietary system, we do not know for sure if Google Translate uses English as a pivot language for Chinese and Arabic. However, even if it does not, we can expect that it would provide reasonable results based on the widespread availability of English parallel corpora (e.g., Linguistic Data Consortium catalog).

[26] For researchers looking to apply these methods to their own texts, we recommend English as a useful default choice for a common language, even if some of the documents are already in English. In particular, circumstances with language groups which are closely related or where excellent parallel text corpora or an available different common language may be more appropriate. The applied researcher can always investigate different options by informally evaluating translation quality by using Google Translate to process a small sample of documents.

[27] For our corpus, the full document translation costs approximately US$450, whereas the term translation was approximately US$10 (both with Google Translate accessed through `translateR`). In general, as of summer 2014, translation with the Google API costs US$20 per 1 million characters of text, so 500,000 characters costs US$10, 2 million characters costs US$40, etc. (more information at https://cloud.google.com/translate/v2/pricing). The Microsoft Translator API operates with a very different cost structure. Users sign up for a monthly plan, which caps the total number of characters that can be translated in a single month. It is free to translate up to 2,000,000 characters per month, US$40 for 4,000,000 characters per month, US$160 for 16,000,000 characters per month, etc. (more information at https://datamarket.azure.com/dataset/bing/microsofttranslator). Note that for both Google and Microsoft, a "character" means an escaped, URL-safe character, so documents written in a language like Chinese often become three to four times longer. However, `translateR` automatically converts the characters to their URL-escaped versions, so users do not need to do so manually.

In our case, the two approaches give somewhat comparable results. We *strongly* caution, though, that this may not be true in general. Fortunately, validation of the translation strategy is relatively straightforward. The natural first check is to verify that topics are not exclusively related to a particular language.[28] If this is not a problem, reading documents highly associated with a particular topic in the native language provides a validation of the translation process. If the documents are largely in agreement with the semantic meaning of the concept as represented in the new language, then the loss of information from the approximate translation procedure is likely acceptable. Analysts should of course be attentive to the way that systematic errors in translation will affect the particular argument that they wish to make and adjust accordingly.

While the complete corpus translation approach is to be preferred in general, the term translation strategy can provide a cost-effective alternative in particular cases. We imagine that this might be particularly useful for early exploratory analyses, which can be used to justify the greater expenditure of complete corpus translation approaches.

4.2.2 Correcting for systematic differences between languages

As discussed in Section 2.3.2, machine translation is not an error-free process. In either of the approaches discussed above, there will be untranslated words, mistranslated words, or words with multiple meanings. As such, words that mean the same thing in the Chinese and Arabic corpus could sometimes map onto different words in English that are synonyms of each other. Just as a native Arabic speaker would speak English differently than a native Chinese speaker, using a vocabulary and sentence structure that most closely maps onto their respective native languages, the "way" in which machine translation interprets each language will be different for the two different corpora. These linguistic differences pose a challenge for topic models. We want to ensure that the topics are uncovering differences in semantic content rather than linguistic idiosyncrasies in describing that content.

As discussed in Section 3.3, the STM allows for this facet of a corpus. Within the STM, we can use a *content* covariate to capture variations in word use attributable to observed covariates. Here we include the document's original language as a content covariate in order to capture linguistic differences in describing a topic. This allows us to effectively marginalize over differences in word rate use that arise due to linguistic differences or errors in translation. For example, the Chinese word for liquor translates into "wine" in Google translate. The Arabic word for liquor, however, translates into "spirits." If there were a "party" topic within our corpus, this would allow both the Chinese and Arabic documents to talk about the party, but the Chinese version of the translation would use wine slightly more and the Arabic translation would use spirits slightly more. Crucially, there is a set of common words that do overlap between the two languages, which allows us to learn that these systematic differences between the languages are related words and not completely separate concepts.[29]

4.2.3 Results

Next, we discuss the results of our illustrative analysis. For all of our analyses, we used a 15-topic model, using an indicator variable for what language community generated the social media post as both topic prevalence and content covariate, as well as a smooth function of time (date of the post) as a topic prevalence covariate. For simplicity, we focus on three different substantive topics in this analysis. The first, which we label "attack," deals with concerns about the United States attacking one's own country or society. The second, labeled "human rights," deals with posts about the

[28] Note that this need not signal a problem, as a topic could actually be specific to a particular country or language. We merely include this to emphasize that such findings should be checked to ensure that they did not arise by a failure in the translation process.

[29] Note the similarity here to the multilingual models discussed in Section 3.2. While those models explicitly maintain models in two or more languages using external alignment information, here we are maintaining a model in only one language but allowing for limited residual variations from the original language. This provides a more parsimonious model structure and facilitates interpretation of the model results. Situations that call for an explicit representation of the topics within multiple languages would be better served by some of the alternatives discussed in Section 3.2.

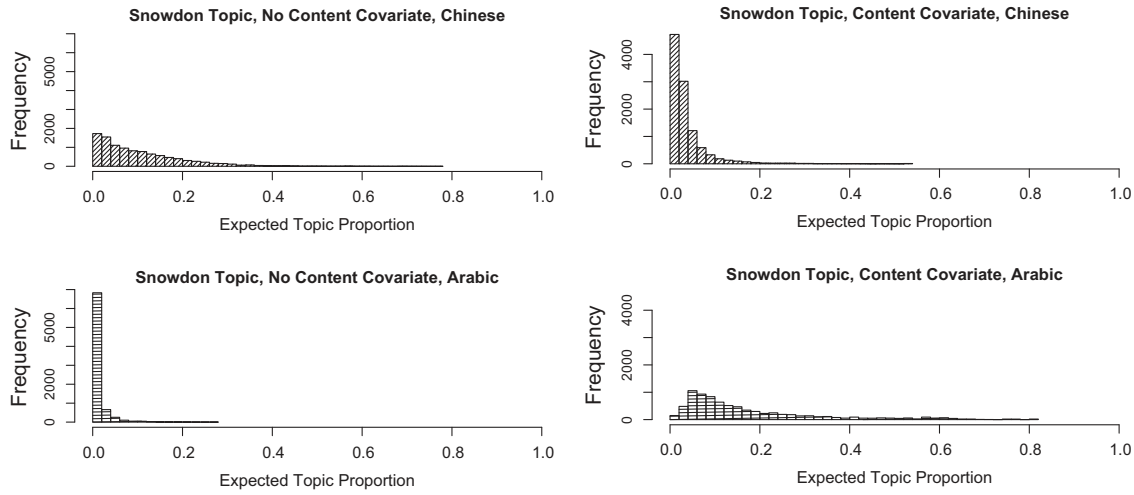


Fig. 5 Histogram of topic proportions for the topic where the word "Snowdon" is most important. Without a content covariate, this topic is dominated by Chinese tweets and has very few Arabic tweets. With a content covariate, this topic mixes between Chinese and Arabic tweets.

implications of the Snowden episode for American credibility on issues related to freedom and human rights. The third, labeled "asylum," concerns news updates about Snowden's movements and whether or not he will be granted asylum and in which country.

First, we emphasize the role of the content covariates in handling the multiple source languages. For some reason, the Chinese version of Snowden's last name translated to "Snowdon," instead of "Snowden," whereas the Chinese version of Snowden's full name translated to "Edward Snowden." This was not the case in Arabic. Therefore, the Chinese examples were likely to use the word "Snowdon," in addition to "Snowden." "Snowdon" did not appear in the Arabic texts at all. Similarly, the Chinese encoding in Google Translate creates the word "quote" when a quotation mark appears. Therefore, many of the words in the Chinese corpus have "quote" attached to them, for example "quotsnowden" or "quotprism."

Of course, the analyst could go through and identify each of these mistakes and correct them, but this would be time-consuming or impossible for larger tasks. By modeling the fact that machine translation will make different mistakes in Chinese than in Arabic using a content covariate, we allow Chinese and Arabic tweets to talk about the same topic, while allowing the tweets from each language to use slightly modified versions of the vocabulary.

Consider the "Snowdon" mistake. For purposes of comparison, we ran a topic model that did not include a content covariate. Within this topic model, the word "Snowdon" pinned down its own topic. Because Snowdon was one of the words defining the topic, it was completely dominated by Chinese tweets; no Arabic tweets were estimated to have more than 0.1 of this topic (see Fig. 5). However, this is a mistake. Chinese tweets translated to "Snowdon" are often discussing the same topic as Arabic tweeters using "Snowden." When we include the content covariate, "Snowdon" appears in a topic with "Snowden," and this topic is similarly distributed in Chinese and Arabic tweets. Had we failed to include a content covariate, we would have created a topic falsely associated with the Chinese tweets.

We now explore the results of the model and compare the full machine translation to the translation of the document-term matrix. To illustrate, we focus first on the two topics related to the image of the United States in the eyes of Chinese- and Arabic-language tweeters, namely the "attack" and "human rights" topics.

The "attack" topic, which discusses the U.S. "attacking" other countries, particularly focused on Snowden's allegations that the U.S. government hacked into Chinese government agencies and businesses. This topic contains words such as "China," "company," "attack," and "relationship." Many of the tweets question the United States–China bilateral relationship going forward. Fig. 6 shows an example of a tweet that is largely devoted to this topic, displaying both the original text

[Snowden broke the news that the American government invasion of China Network] Snowden published evidence for many years, said that the American government intrusion Chinese network has at least four years, the goal of the US government to reach hundreds of hackers, which also includes schools. Hackers typically by a massive invasion through the router, then the invasion of thousands of computers in one fell swoop, no one individual computer intrusion. http://t.cn/zHRpotF	【斯诺登爆料称美国政府入侵中国网络多年】斯诺登公布证据，表示美国政府网络入侵中国网络至少有四年时间，美国政府黑客攻击的目标达到上百个，其中还包括学校。黑客的方式通常是透过入侵巨型的路由器，然后一举入侵成千上万的电脑，无需一一入侵个别电脑。 http://t.cn/zHRpotF

Fig. 6 Example post for the attack topic.

Logically, especially from the Western approach to human rights warriors from other countries show a similar point of view, the US-led West should pay tribute to Snowden, gratitude, and ultimately awarded the nomination Hasa Rove European Human Rights Award, or simply America's own Herman - Hammett Human Rights Award, Lantos Human Rights Award, it is not the British Parliament as well as the first Westminster Human Rights Award. http://t.cn/zH3iwJ6	从逻辑上讲，特别是从西方对待来自其他国家类似的人权勇士的表现来看，以美国为首的西方应该向斯诺登表达敬意、谢意，提名并最终颁发欧洲的哈萨罗夫人权奖、或者干脆美国自己的赫尔曼-哈米特人权奖、兰托斯人权奖，实在不行还有英国议会首威斯敏斯特人权奖。http://t.cn/zH3iwJ6

Fig. 7 Example post for the human rights topic.

and its translation. We generated the translated text by calling the Google Translate API with `translateR`. Note also that the translation captures the essence of the post.

The "human rights" topic discusses the U.S. record on human rights and whether the Snowden disclosures undermine this record. This topic contains words such as "violate," "freedom," "human," "right," and "traitor." Some of the posts also discuss whether the United States is a hypocrite, violating U.S. citizens' human rights while also advocating for greater human rights protection abroad. Fig. 7 displays a tweet on this topic, where again we use `translateR` to access the Google Translate API for the translated text.

Given the dramatic cost difference between the full-text and DTM translations, it is useful to investigate the similarity of the resulting topic models. If the DTM translation produces comparable results, it will clearly be preferable on cost alone. We investigate the similarity of the models for our two topics of interest, cautioning that congruence between the models for this case does not produce a general result.

We examine the alignment between models by comparing the topic-word distributions of all topics in both models. Because the two models use different vocabularies, we identify the common

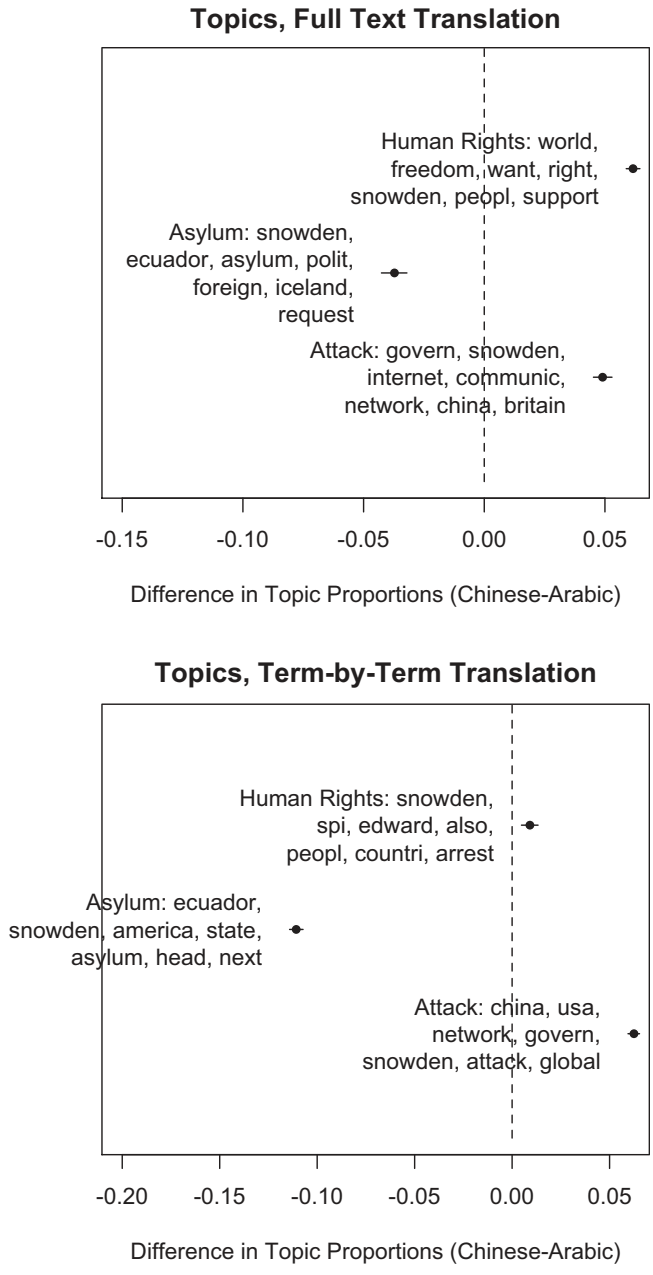


Fig. 8 Topics related to U.S. reputation. The top plot is the estimation with full-text translation, and the bottom plot is the estimation of these topics with DTM-translation. Both plots show the relationship between the topics and the Chinese and Arabic corpuses.

terms and calculate the correlation between every pair of topics using the overlapping words.[30] Some of the topics align quite clearly, including the two we have highlighted above. In Online Appendix F.4, we provide a visualization of the correlations between all topic pairs.

We explore the question of model alignment further by investigating how our aggregate inferences about the relative rates of topical prevalence would change under the different

[30]Specifically, we construct a marginal estimate of the topic word distribution β by weighting the Chinese- and Arabic-specific version of the topics by their relative frequency in the corpus. We then take the intersection of the vocabulary between the two models and calculate the correlation between the distributions over those words for each topic pair.

translation strategies. In Fig. 8, we plot the three topics and their estimates under each of the two translation methods. Note that the three displayed topics—"Attack," "Human Rights," and "Asylum"—all have similar frequent terms and substantively similar estimates. For both the full-text translation and the document-term matrix translation, the "Attack" and "Human Rights" topics are more associated with Chinese posts than with Arabic posts. At least in this case, two investigators using different translation methods might have reached similar substantive conclusions, namely that microbloggers in China seem very ready to condemn the U.S. government for hacking Chinese companies and the government and for "trampling" human rights. This analysis is further explored in Online Appendix F, along with an overview of additional topics in the model and the technical details of the estimation.

Which topics are more associated with Arabic tweets? Arabic tweeters are more likely to be sharing news about the Snowden disclosures. The "Asylum" topic is associated with Arabic tweets and is related to speculation about where Edward Snowden will end up seeking asylum. This topic contains words such as "Ecuador," "Iceland," "shelter," "request," and "asylum." Arabic tweeters are much more likely to be sharing news, rather than opinions about the U.S. government's reputation.

These results begin to speak to our original interest in the ways that the reputation of the United States was damaged in the eyes of Chinese and Middle Eastern social media users during the Snowden incident. However, we also find that these topics were more prevalent within the Chinese corpus than the Arabic corpus. This is unlike other events where there are strong reactions to U.S. intervention in the Middle East by Arabic twitter users (Jamal et al. n.d.). The Snowden disclosures seemed to affect Chinese perceptions of the United States more strongly; not only was there considerable outcry about U.S. cyber-intervention in China, but the Snowden event generated discussion of how the United States in fact opposes human rights and is less democratic than the U.S. attempts to seem on the world stage. Perhaps, given the perception that U.S. cyber activities targeted China, the Chinese response is consistent with previous work focusing on the Middle East. Using this type of workflow, scholars could examine reactions by many countries to other world events that the United States is involved in.

5 Conclusion

The volume of textual data is growing rapidly throughout the world. The form of this textual data is no longer simply in the form of newspapers, books, etc., but also in social media and other internet-based content that puts even fewer restrictions on the generation of textual data (e.g., Barberá 2012). There is no sign that this trend will change. Even if a tiny fraction of these data is ultimately of interest to comparativists, they will need to understand a range of issues relevant to different languages that are actively being studied by scholars.

This article introduces comparativists to a range of important topics in textual analysis. We walked through a variety of research questions that comparative politics scholars have been asking and answering with textual data, and introduced the basics of textual processing with a focus on non-English texts. Next, we discussed the managing and preprocessing of text from a multilanguage perspective, including a brief discussion of new software such as `txtorg`, as well as a discussion of machine-based translation where we introduce a new `R` package `translateR` that provides easy access to the Google Translate API. Next, we briefly discussed techniques for text analysis, emphasizing the existing tools for multilingual text analysis. Finally, we used the STM to provide two examples of how comparativists can use metadata to incorporate their knowledge of corpus structure into unsupervised learning, including a novel way to use the STM model when text has first been translated to a single language.

Future developments designed to address remaining challenges could proceed in a number of different directions. We are particularly interested in harnessing the ever-increasing advances in automated translation with existing text analysis techniques. No doubt existing translation methods are imperfect; however, translation is an active research area in academia and industry, which suggests that these systems will continue to improve over time. An open question for social scientists is how to best leverage these developments for applied research. A critical part of this

process is developing diagnostic tools for assessing the sensitivity of our analysis tools to translation error.

Finally, we plan to continue developing open-source software which brings the necessary tools for automated text analysis to the end-user. The three software packages described here cover different portions of the text analysis workflow, from processing of texts to estimating the model. We plan to continue refining these tools with comparative politics scholars in mind, while developing new software, including a browser-based system for interactive topic model exploration.

Funding

Dustin Tingley gratefully acknowledges his Dean's support for this project.

References

Alfonseca, E., S. Bilac, and S. Pharies. 2008. Decompounding query keywords from compounding languages. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, 253–256. Association for Computational Linguistics.

Barberá, P. 2012. Birds of the same feather tweet together: Bayesian ideal point estimation using twitter data. In *APSA 2012 Annual Meeting Paper*.

Baturo, A., and S. Mikhaylov. 2013. Life of Brian revisited: Assessing informational and non-informational leadership tools. *Political Science Research and Methods* 1(01):139–57.

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

Blei, D. M., and J. D. Lafferty. 2007. A correlated topic model of science. *Annals of Applied Statistics* 1(1):17–35.

Boyd-Graber, J., and D. M. Blei. 2009. Multilingual topic models for unaligned text. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pp. 75–82. AUAI Press.

Boyd-Graber, J., and P. Resnik. 2010. Holistic sentiment analysis across languages: Multilingual supervised latent dirichlet allocation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 45–55. Association for Computational Linguistics.

Brachman, J. 2009. *Global Jihadism*. New York: Routledge.

Brady, H. E., and D. Collier. 2010. *Rethinking social inquiry: Diverse tools, shared standards*. Lanham, MD: Rowman & Littlefield.

Brown, P. F., J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics* 16(2):79–85.

Brown, P. F., V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics* 19(2):263–311.

Budge, I., K. Hans-Dieter, V. Andrea, B. Judith, and T. Eric. 2001. *Mapping Policy Preferences: Estimates for Parties, Electors, and Governments 1945–1998*. Oxford: Oxford University Press, Oxford, UK.

Campbell, R. S., and J. W. Pennebaker. 2003. The secret life of pronouns flexibility in writing style and physical health. *Psychological Science* 14(1):60–65.

Catalinac, A. 2014. Pork to policy: The Rise of National Security in Elections in Japan, unpublished manuscript.

Cheng, K.-S., G. H. Young, and K.-F. Wong. 1999. A study on word-based and integral-bit Chinese text compression algorithms. *Journal of the American Society for Information Science* 50(3):218–28.

Chiozza, G. 2009. *Anti-Americanism and the American world order*. Baltimore: Johns Hopkins University Press.

Coscia, M., and V. Rios. 2012. Knowing where and how criminal organizations operate using web content. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 1412–1421. ACM.

Eggers, A., and A. Spirling 2011. Partisan convergence in executive-legislative interactions modeling debates in the House of Commons, 1832–1915. unpublished manuscript.

Farrell, H., and M. Finnemore 2013. The end of hypocrisy: American foreign policy in the age of leaks. *Foreign Affairs* 92:22.

Feinerer, I., K. Hornik, and D. Meyer. 2008. Text mining infrastructure in R. *Journal of Statistical Software* 25(5):1–54.

Fokkens, A., M. Van Erp, M. Postma, T. Pedersen, P. Vossen, and N. Freire. 2013. Offspring from reproduction problems: What replication failure teaches us. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1691–1701, Sofia, Bulgaria, August. Association for Computational Linguistics.

George, A., and A. Bennett. 2005. *Case studies and theory development in the social sciences*. Cambridge, MA: MIT Press.

Griffiths, T. L., and M. Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America* 101(Suppl 1):5228–235.

Grimmer, J. 2010. A Bayesian hierarchical topic model for political texts: Measuring expressed agendas in Senate press releases. *Political Analysis* 18(1):1.

Grimmer, J., and B. M. Stewart. 2013. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis* 21(3):267–97.

Habash, N., and J. Hu. 2009. Improving Arabic-Chinese statistical machine translation using English as pivot language. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pp. 173–81. Association for Computational Linguistics.

Harman, D. 1991. How effective is suffixing? *JASIS* 42(1):7–15.

Hollink, V., J. Kamps, C. Monz, and M. De Rijke. 2004. Monolingual document retrieval for European languages. *Information Retrieval* 7(1–2):33–52.

Hu, Y., K. Zhai, V. Eidelman, and J. Boyd-Graber. 2014. Polylingual tree-based topic models for translation domain adaptation. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers): 1166–1176.

Hull, D. A. 1996. Stemming algorithms: A case study for detailed evaluation. *JASIS* 47(1):70–84.

Jamal, A., R. O. Keohane, D. Romney, and D. Tingley. n.d. Anti-Americanism or anti-interventionism? Evidence from the Arabic Twitter universe. *Perspectives on Politics*. Forthcoming.

Katzenstein, P. J., and R. O. Keohane. 2007. Varieties of anti-Americanism: A framework for analysis. In *Anti-Americanisms in world politics*, eds. P. J. Katzenstein and R. O. Keohane, 9–38. Ithaca: Cornell University Press.

King, G., J. Pan, and M. E. Roberts. 2013. How censorship in China allows government criticism but silences collective expression. *American Political Science Review* 107:1–18.

Koehn, P. 2009. *Statistical machine translation*. Cambridge, UK: Cambridge University Press.

Krovetz, R. J. 1995. Word-sense disambiguation for large text databases PhD thesis, University of Massachusetts, Amherst.

Laver, M., K. Benoit, and J. Garry. 2003. Extracting policy positions from political texts using words as data. *American Political Science Review* 97(02):311–31.

Lunde, K. 2009. *CJKV information processing*. New York, NY: O'Reilly Media, Inc.

Lynch, M. 2007. Anti-Americanism in the Arab world. In *Anti-Americanisms in world politics*, eds. P. J. Katzenstein and R. O. Keohane, 196–224. Ithaca: Cornell University Press.

Manning, C. D., P. Raghavan, and H. Schütze. 2008. *Introduction to information retrieval*, Vol. 1. Cambridge: Cambridge University Press.

McCallum, A. K. 2002. Mallet: A machine learning for language toolkit. Available at http://mallet.cs.umass.edu.

McCants, W. 2006. Militant ideology atlas. Technical report, Combating Terrorism Center, U.S. Military Academy.

Miller, M. C. 2013. *Wronged by empire: Post-imperial ideology and foreign policy in India and China*. Stanford, CA: Stanford University Press.

Mimno, D., H. M. Wallach, J. Naradowsky, D. A. Smith, and A. McCallum. 2009. Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2*, 880–889. Association for Computational Linguistics.

Mosteller, F., and D. L. Wallace. 1963. Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed Federalist Papers. *Journal of the American Statistical Association* 58(302):275–309.

Nielsen, R. 2013. The lonely Jihadist: Weak networks and the radicalization of Muslim clerics. PhD Thesis, Harvard University. Ann Arbor: ProQuest/UMI (Publication No. 3567018).

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of Association for Computational Linguistics*, 311–318. Association for Computational Linguistics.

Paul, M., H. Yamamoto, E. Sumita, and S. Nakamura. 2009. On the importance of pivot language selection for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pp. 221–224. Association for Computational Linguistics.

Quinn, K., B. Monroe, M. Colaresi, M. Crespin, and D. Radev. 2010. How to analyze political attention with minimal assumptions and costs. *American Journal of Political Science* 54(1):209–228.

Roberts, M. E., B. M. Stewart, and E. Airoldi. 2015. A model of text for experimentation in the social sciences. Unpublished manuscript.

Roberts, M. E., B. M. Stewart, and D. Tingley. 2014. *stm: R package for structural topic models*. R package version 0.6.21. software package http://structuraltopicmodel.com/.

Roberts, M. E., B. M. Stewart, D. Tingley, and E. M. Airoldi. 2013. The structural topic model and applied social science. *Advances in Neural Information Processing Systems Workshop on Topic Models: Computation, Application, and Evaluation*.

Roberts, M. E., B. M. Stewart, D. Tingley, C. Lucas, J. Leder-Luis, S. Gadarian, B. Albertson, and D. Rand. 2014. Structural topic models for open-ended survey responses. *American Journal of Political Science* 58(4):1064–1082.

Rubin, B. 2002. The real roots of Arab anti-Americanism. *Foreign Affairs* 81(6):73–85.

Salton, G. 1989. *Automatic text processing: The transformation, analysis, and retrieval of information by computer*. Boston, MA: Addison-Wesley.

Schonhardt-Bailey, C. 2006. *From the Corn Laws to free trade [electronic resource]: Interests, ideas, and institutions in historical perspective*. Cambridge, MA: MIT Press.

Schrodt, P. A., and D. J. Gerner. 1994. Validity assessment of a machine-coded event data set for the Middle East, 1982–92. *American Journal of Political Science* 38(3):825–854.

Slapin, J. B., and S.-O. Proksch. 2008. A scaling model for estimating time-series party positions from texts. *American Journal of Political Science* 52(3):705–722.

Stewart, B. M., and Y. M. Zhukov. 2009. Use of force and civil–military relations in Russia: An automated content analysis. *Small Wars & Insurgencies* 20(2):319–343.

Stockmann, D. 2012. *Media commercialization and authoritarian rule in China*. New York, NY: Cambridge University Press.

Telhami, S. 2002. *The stakes: America and the Middle East*. Boulder, CO: Westview Press.

Tseng, H., P. Chang, G. Andrew, D. Jurafsky, and C. Manning. 2005. A conditional random field word segmenter for Sighan Bakeoff 2005. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, Vol. 171. Jeju Island, Korea.

Utiyama, M., and H. Isahara. 2007. A comparison of pivot methods for phrase-based statistical machine translation. *In 2007 Proceedings of NAACL/HLT*, pp. 484–491.

Van Atteveldt, W., J. Kleinnijenhuis, and N. Ruigrok. 2008. Parsing, semantic networks, and political authority using syntactic analysis to extract semantic relations from Dutch newspaper articles. *Political Analysis* 16(4):428–446.

Volkens, A., P. Lehmann, N. Merz, S. Regel, A. Werner, O. Lacewell, and H. Schultze. 2013. The manifesto data collection. In *Manifesto Project (MRG/CMP/MARPOR)*. Berlin: Wissenschaftszentrum Berlin für Sozialforschung (WZB).

Zhao, B., and E. P. Xing. 2006. Bitam: Bilingual topic admixture models for word alignment. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, pp. 969–76. Association for Computational Linguistics.

A Online Appendix: Additional Text Preparation Details

A.1 Conversion to Digital Format

Text gathered from a variety of different sources may not be immediately readable by a computer. First, the data itself might be pictures of text taken from an archive or hand-written manuscripts that are not yet digitized. In these cases, Optical Character Recognition (OCR) technologies may be required.[1] Even if they have been digitized, texts may be stored digitally using different encodings and therefore not immediately usable as one corpus.

A.2 Bag of words

The bag-of-words assumption underlies many statistical models due to its relative simplicity. The simplest method of expanding beyond this approach is to allow each item in the dictionary to represent a multi-word phrase rather than a single word. Because the space of ordered word pairs (bigrams) and ordered word triples (trigrams) is significantly higher than using a single word, this procedure is often coupled with some method of selecting the most relevant phrases (Jensen et al., 2012; Gentzkow and Shapiro, 2010). This process can often engender subtle difficulties in interpretation (see for example the critique of trigrams in Spirling (2012)). More complex approaches have included the use of string kernels (Spirling, 2011) and word transition distributions (Wallach, 2006). We choose to focus here on the simple bag-of-words model but emphasize that vocabulary can be manually modified to include contextually appropriate phrases.

A.3 Additional pre-processing options

As is typically the case in text analysis, the illustrations presented in this paper required a good deal of preprocessing. In determining how to preprocess our data for each of these examples, we drew on substantive knowledge of the relevant language, cases, and questions, and suggest that others do the same. But because no analysis is the same, researchers may need to preprocess their data in ways not presented in this paper. For example, users may wish to correct spelling errors, especially when analyzing survey responses or social media posts where such errors are common. This is easily accomplished in `R` with the built-in `aspell()` function. Users may also wish to expand contractions (if they are not stemming), to remove punctuation, to make text lowercase (except for

[1] OCR works by using pattern recognition to identify patterns within the image file that look like characters and transfer them into machine-encoded text. Some OCR software are dictionary-based, searching for word within the dictionary that looks most like the image. While OCR software is rarely, if ever, 100% accurate, clearly-written texts can often achieve accuracy rates of above 90%, which is enough to understand the content of the text and to use automated content analysis. Even with high error rates, errors are unlikely to be correlated with most quantities of interest, so the measurement error can be corrected for. There are many open-source OCR software packages, (see for example FreeOCR: `http://www.free-ocr.com/` or Tesseract: `https://code.google.com/p/tesseract-ocr/`) and we recommend that users try out several packages on a given text, as none are perfect, but some may work better with particular image files. If OCR is not possible it is often possible to pay for data entry.

proper nouns), etc., most of which can be accomplished easily with `tm` or `txtorg`.

B Online Appendix: A Review for Text Analysis Techniques for Comparativists

In this online appendix, we provide a brief pedagogical overview of supervised and unsupervised methods with an emphasis on choosing the best approach for applied research. We refer interested readers to Grimmer and Stewart (2013) for more detailed discussion on particular models, but here we focus more on examples to help applied comparative politics researchers. We emphasize two aspects of analysis throughout: the processes by which the analyst incorporates information into the model and the importance of validation, by which we mean checking model results with substantive knowledge and reference to the original text.

Oftentimes the virtues and limitations of methods are best illustrated through a familiar set of documents. In the next few sections we will illustrate methods with toy examples using a corpus comprised of the last 6 decades of research articles from the *American Political Science Review* containing the phrase "comparative politics". We obtain the word count information from JSTOR's Data-For-Research site, selecting articles from 1950-2012. We limit the classification to research articles and choose only articles in excess of 5 pages (in an attempt to further remove reviews, rejoinders etc.), yielding 417 articles. These texts are accompanied by a variety of metadata elements including the authors, titles and year of publication. At each stage we point the reader to applied examples of the methods we discuss.

B.1 Supervised Methods

In their most basic form, supervised methods are a way of replicating work done by the human analyst on a small scale, to a much larger set of documents. This allows the analyst to undertake an analysis that one could imagine a human performing with infinite time but would for all practical purposes be intractable. We start with simple word counting approaches and their natural extension, document scaling by weighted word counts.

Keywords Methods and Document Scaling The simplest form of supervision is counting human-selected keywords. The choice to focus on the counts of particular chosen words is itself a case of a very simple supervised model. When the quantity of interest is precisely represented, keywords can be a conceptually and computationally simple approach. Some approaches (such as Yoshikoder)[2] assist the user in interpreting the text by placing keywords within "context" (showing example sentences and phrases which contain the keyword). These methods are a helpful validation method but they still assume that the use of a particular word is the same across different documents regardless of the context in which it is used.

Figure 1 shows a simple keyword trend tracking the use of the word "causal" in *APSR*. This accords with our understanding of a general rise in interest in causal inference methods over the last 6 decades. While informative, it is important to emphasize that

[2] `http://conjugateprior.org/software/`

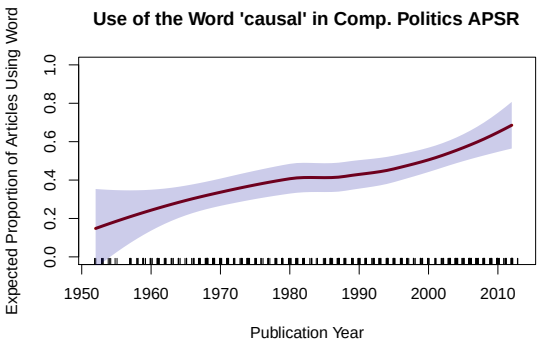


Figure 1: Illustration of Simple Keyword trends. This shows the rise in the proportion of articles in *APSR* using the word 'causal.' This is a suggestive trend which accords with our understanding of the literature but requires additional supervision from the researcher to determine how much we can infer. The plot shows the Loess-smoothed estimate of the expected proportion of articles using the word "causal" with 95% confidence intervals.

this need not indicate that researchers are using causal inference methods or even that the use of the word is consistent over the time frame. For that we would need to turn to more complex methods.

A common extension to counting methods is to consider weighted counts. These methods are most often applied to sentiment analysis where dictionaries encode the valence of individual words, for example, "angry" might get a more negative sentiment score than "annoyed." There are dictionaries available for affect, cognitive mechanisms, sentiment and various topics (Stone et al., 1966; Pennebaker et al., 2001). However, these dictionaries were developed for particular types of texts and may not extend well to other domains (Grimmer and Stewart, 2013).

Dictionaries of weighted terms can also be learned automatically from human annotated task. In this setting, the user assigns texts to (generally extreme ends of) categories and uses a model to determine words that are most "distinctive" or predictive of that class. Words that have high use under one class but low use under another are given higher scores. Care must be taken with the interpretation of continuous scores generated from these methods. The continuous nature is (generally) derived from the predictive power of the features under a discrete classification system. This need not always correspond to an intuitive notion of intensity (e.g. "more conservative") although it does work well in many cases.[3]

To emphasize this point we give an example from the *APSR* corpus. Here we leverage the authorship metadata to group documents into those which are co-authored and those which are solo-authored. In order to characterize what is distinctive about the co-authored articles we create a "co-author" score where we scale higher scores that indicate that the

[3]Monroe et al. (2008) provide a comparison of various weighting methods of words for the purposes of feature selection.

words are more predictive of co-authorship.[4] We give the top 10 words for each side in Table 1. The results suggest that quantitative empirical work is often co-authored

	Solo Words	Score	Coauthor Words	Score
1	rearmament	-5.061	ethnopolitical	5.259
2	lebanese	-4.846	elf	4.212
3	theology	-4.788	civics	4.208
4	machiavelli	-4.635	humphrey	4.078
5	crops	-4.514	contemporaneously	3.952
6	loc	-4.444	rd	3.95
7	hitler	-4.444	AY	3.866
8	colonization	-4.444	pie	3.862
9	cdu	-4.444	supervisors	3.845
10	mussolini	-4.42	admissible	3.786

Table 1: Top words indicating coauthored or solo-authored work with scaled values. Words with higher absolute value scores have a stronger connection.

whereas historical and qualitative work is more likely to be solo-authored. The words don't necessarily capture a theoretical conception of "collaborativeness"; they are simply the most predictive of co-authorship.

This example affords us the opportunity to emphasize the importance of validation in computer-assisted text methods. For readers of comparative politics it may be easy to look at the set of coauthor words and tell a consistent story which implies a particular contextual meaning for each word; "elf" refers to the ethno-linguistic fractionalization index, "rd" refers to regression discontinuity estimators etc. "Humphrey" the reader may naturally assume refers to Macartan Humphreys, as his research is often quantitative and is often coauthored. However, examination of the documents in our sample which contain the word "humphrey" reveals that the finding is driven largely by a few outliers. One article which uses "humphrey" an astonishing 55 times is titled "Continuity and Change in American Politics: Parties and Issues in the 1968 Election" (Converse et al., 1969). The article discusses Vice President Hubert Humphrey, a subject which clearly has a quite different interpretation. This example illustrates the need for checking model interpretation against the original texts even when a collection of words appear to tell a very clear story.

Example Applications of Keyword and Scaling Methods Several good examples of keyword methods can already be found in the comparative politics literature. Johnston and Stockmann (2007) use stories from the *China Daily* to study Chinese attitudes toward Americans immediately after the 2004 tsunami in South Asia. They identify positive and negative terms surrounding references to the United States in order to measure attitudes toward American relief efforts following the tsunami. Stockmann (2011) analyzes how media marketization in China influences views toward the United States. She finds that after extensive media marketization in the early 2000s, Chinese newspapers use more negative words surrounding the United States than positive.

[4]Specifically we use Taddy (2013)'s inverse regression procedure which fits a regularized multinomial logistic regression with words as the output and the co-authorship indicator as the predictor. Scores are essentially the regularized log-odds of word use.

Scaling is the most widely used automated text analysis within comparative politics due to its prominent role in the study of partisan ideology. Perhaps the most prominent example is the WordScores method (Laver et al., 2003) for analyzing ideology in the Comparative Manifestos Project. As this literature is quite well developed in comparative politics we do not dwell on it at length. For more methodological details on the connections between scaling methods and supervised learning we refer readers to Lowe (2008); Beauchamp (Beauchamp); Taddy (2013).

Classification/Prediction Document classification is perhaps the most common form of supervised learning. In this setting researchers develop a categorization by hand and use it to label a random subset of the documents which we call the 'training set.' The model learns a set of parameters from the training set which it uses to assign the remaining documents into our original categories. Classification is most useful in cases where (1) discrete individual decisions need to be made or (2) where an extensive coding system already exists.[5] There are an enormous set of possible classification algorithms that the analyst can employ, but their common structure makes it possible to provide general advice that applies to nearly the entire set.

In order to provide some context, we outline an example workflow for document classification. The researcher must first develop a categorization scheme which assign each document to a category in a way that is both mutually exclusive and exhaustive. Best practice is to develop a codebook which provides sufficient detail that an informed third party would be able to assign a new document within the existing scheme (Krippendorff, 2012). The researcher would ideally want all of the documents hand-coded into this categorization scheme, but with a large number documents, hand-coding each individual document would be too time-consuming and expensive. Instead, we hand code only a sample, with the algorithm classifying the remaining unseen documents.

To teach the algorithm how to classify, researchers select a (preferably random) sample of documents to be manually coded by human coders into the categories. In order to ensure that the categories are consistent across coders, each document is usually coded by two or more researchers (who are provided with the aforementioned codebook) with a final arbiter comparing the coding to measure the "inter-coder reliability", or how consistent the coding is between researchers. In practice this is often an iterative process of developing the coding scheme and refining the codebook. Once the coders have achieved a high inter-coder reliability, the resulting "training set" of coded documents and the remainder of the uncoded documents are then given to the classifier as a term-document matrix. Using this representation the algorithm learns the parameters for a model which can classify unseen documents. This classifier is then applied to the remainder of the corpus.

A significant advantage of supervised document classification is the relative ease in evaluating machine performance. The accuracy of the classifier can be checked by manually inspecting new documents and comparing the classifier's coding to manual cod-

[5]The canonical example of the first case is spam filtering, where your email system must decide for each email if it is spam or regular. The Congressional Bills Project which classifies documents according to the topic system developed by the Policy Agendas Project is an example of the second case (Hillard et al., 2008).

ing. With a sufficiently large training set classifier accuracy can be assessed using cross-validation (Grimmer and Stewart, 2013).

Although it is relatively easy to assess individual document codings, it is often beneficial, although significantly harder, to assess the calibration of the model's predictions. For example imagine a setting where we classify emails into 'spam' or 'not spam.' Assessing whether an individual item marked 'spam' is correctly identified is often trivially easy. It takes significantly more data to assess whether items marked as 70% likely to be spam are actually spam 70% of the time in expectation. While many classification algorithms tend to make strong individual predictions, these predicted probabilities are often overly confident.

Calibration of the model becomes particularly important when we are interested in making inference about a group of documents. If, for example, we wanted to assess what proportion of *APSR* articles in our corpus are about 'institutions' we could train a classifier and then sum up the predicted probability that each document falls into the 'institutions' category. However, if our classifier is poorly calibrated, this estimate of the group proportions can be severely biased.

One method developed within political science, `ReadMe`, leverages this particular goal to provide more accurate estimates of category proportions than classifiers focused on individual classifications (Hopkins and King, 2010). `ReadMe` returns only the group category proportions and does not provide individual document level categorizations. This focus on estimating the population proportion correctly allows for increased accuracy but makes it inappropriate for settings where the classification of each individual document must be known.

The workflow for using `ReadMe` is substantially similar to individual document classification. The lack of individual document assignments can mildly complicate the process of validation, but ultimately it is no more difficult than attempting to assess the calibration of classification probabilities in the individual classifier setting. The entire workflow for using `ReadMe` is summarized in the user manual for the software (Hopkins et al., 2010) with much of the advice also being applicable to the broader set of classification algorithms.

Example Applications of Document Classification King et al. (2013) provide one recent example of using `ReadMe` in comparative politics. King et al. (2013) download millions of blogposts before the Chinese government is able to censor them, then return to the blog posts later to see whether they were censored, providing one of the first large-scale measurements of government censorship in China. In one part of the paper, the authors use `ReadMe` to test whether censorship in China is focused on removing blog posts about collective action events or removing blog posts with criticism of the state. Jamal et al. (nd) provide another example. They examine views of America that are being expressed in Arabic and on Twitter. Rather than relying on public opinion data to understand views of America, Jamal et al. (nd) innovate by analyzing millions of tweets about America. They also create specific categories to analyze responses to events, like the Boston Marathon bombing.

When and How to Use It Supervised methods are best used when the analyst is looking to identify a particular quantity within the text. In the case of document classi-

fication errors within the algorithm can be straightforwardly assessed by comparing the algorithm's decisions to the human-created codebook.[6] For keyword methods using pre-constructed dictionaries of words be sure to validate the resulting methods to insure that they represent the intended concept.

The software tools for document scaling and classification are particularly well developed. Within the `R` language, there are numerous packages including `RTextTools` (Jurka et al., 2013) for classification and `austin` for scaling (Lowe, 2013).[7]

B.2 Unsupervised Methods

Where supervised methods amplify human effort by attempting to replicate human effort expended at the beginning of the process, unsupervised methods suggest new ways of organizing texts. This in turn means that human effort is primarily focused on the interpretation of the results. Thus these methods are useful when seeking to broadly characterize what a corpus of texts is about without strong *a priori* assumptions about what that might mean. While originally used primarily for exploratory analysis, increasing attention has been paid to using these methods in the context of measurement, generally in contexts where the development of a coding system would be prohibitively expensive (Quinn et al., 2010; Grimmer, 2010) .

Keywords and Scaling As in the supervised case we can extend the word frequency approach to weighted word frequencies. We posit a continuous latent variable which explains the text such that each document is assigned to lie along a continuum. This is the approach taken in the Wordfish algorithm for studying political ideology (Slapin and Proksch, 2008) and is the unsupervised analog of our analysis with the coauthorship variable used previously. [8]

Document Clustering Document clustering assumes that each document belongs to a single latent group, and this group provides the best explanation for word use. This approach has been used in political science for analyzing the contents of Senate press release (Grimmer, 2010) and speech on the floor of the House (Quinn et al., 2010). Clusters of documents can either each be distinct or nested into hierarchies themselves.

For longer and more complex documents (e.g. our *APSR* corpus) the assumption that each document falls within only a single cluster is too restrictive. Mixed-membership models, such as Latent Dirichlet Allocation (LDA) (Blei, 2012), assume that each word comes from a single topic, and that each document comes from a mixture of topics. Thus, a document is represented as the proportion of its words that come from each topic.[9]

[6]Specifically we often validate accuracy via cross-validation. See Grimmer and Stewart (2013) for details.

[7]Documentation on `RTextTools` can be found at `http://www.rtexttools.com/`. Documentation for `austin` can be found urlhttp://conjugateprior.org/software/austin/. Both have an excellent set of examples for getting started.

[8]Of course there is no guarantee that the recovered dimension will correspond to any specific political concept (Grimmer and Stewart, 2013). In general, unsupervised scaling methods will characterize the dominant source of variation in the texts whether that variation is topical, ideological or stylistic. For other examples of unsupervised scaling methods, see Elff (2013).

[9]To help distinguish the two types of models we use the term "cluster" when discussing single mem-

For our APSR corpus we run an LDA model with six topics and in Table 2 display seven informative words which characterize each topic.

1	press,democratic,international,democracy,countries,variables,institutions
2	elections,election,local,members,models,population,levels
3	military,war,conflict,civil,people,leaders,empirical
4	groups,group,interest,problems,foreign,law,latin
5	behavior,cases,society,soviet,values,issue,approach
6	party,parties,model,electoral,vote,voting,effects

Table 2: Top words for a 6 topic LDA model on the APSR corpus.

We emphasize that these topics are discovered from the texts using the patterns of word co-occurrences and are not assumed by the researcher. To give a concrete example, a document mostly coming from topic 2 (about elections and local politics) is "French Local Politics: A Statistical Examination of Grass Roots Consensus" (Kesselman, 1966) and it also draws from the 5th topic on behavior and the 6th topic on parties and voting.

When and How to Use Unsupervised Text Models Unsupervised learning is best applied when the analyst is interested in the contents of a corpus but do not have strong expectations for its structure. Generally all that is necessary to perform document clustering or topic modeling is the algorithm of choice and an assumption about the number of latent groups to be estimated. There is no "correct" specification in a general sense, but different approaches will provide different types of structure. For example specifying the number of clusters or topics provides views of the data at different levels of granularity.

Unsupervised scaling using the Wordfish algorithm (Slapin and Proksch, 2008) is available in the `austin` package (Lowe, 2013). The basic LDA model is implemented in several R packages including `lda` (Chang, 2012) and `topicmodels` (Grün and Hornik, 2011).

Choosing a Strategy When choosing a method for automated text analysis the key is to focus on the best use of human effort. At an abstract level the distinction between unsupervised and supervised learning is extremely clear: supervised learning is learning by human-provided example whereas in unsupervised learning those examples are discovered from the data. In practice the line between the two approaches is somewhat hazier. Nearly all good coding schemes are developed by a combination of *a priori* theory and iterative inspection of the data. Thus even in supervised models, there is a "concept discovery" process. With supervised learning this comes through model fitting and checking errors, in unsupervised learning the concept discovery is simply part of the model.

How then should we think about the difference between the two approaches in applied work? Supervised learning offers a greater degree of control: concepts are completely and exclusively enumerated. This is a powerful assumption which tends to make models easier to estimate and results easier to check (because the concept of an error is very well defined). When the space of possible concepts is easy to define this is considerably more tractable.

bership models and "topic" when describing "mixed membership models."

The choice between supervised and unsupervised learning is subtle in the general case but often is quite obvious in practice. No matter what you choose it is important to validate your results. This helps to prevent conceptual slippage between the statistical model of the text and the theoretical concept which we are claiming to measure. For approaches to validation see Grimmer and Stewart (2013); Quinn et al. (2010); Lowe and Benoit (2013).

C Online Appendix: Machine Translation via translateR

As discussed in the paper great strides have been made in machine translation researcher, especially by large corporations like Google and Microsoft. This paper makes use of a new `R` package `translateR` to help researchers translate multilingual corpora into a common language. `translateR` provides bindings for the Google and Microsoft Translation APIs and can export translated text in a format that easily works with the `stm` package or other text analysis packages.[10] `translateR` has one workhorse function, `translate()`, through which it provides access to both supported APIs. `translate()` takes text in two formats. First, users can provide as input a dataframe where one column is text and the remaining are metadata. In this case, `translate()` returns the original dataframe with the translated text bound to the original data set. This is important for use with `stm`, as the metadata can then be used as covariate information to estimate a structural topic model for the translated text. Second, users can provide text in the form of a character vector. This is valuable for users translating a document-term matrix. The column names of a document-term matrix (i.e., the terms) can be passed to `translate()` as a vector. The translated vector can then be used to rename the columns, thus "translating" the DTM from one language into another.

`translateR` also provides a number of "sanity checks" to protect users. For instance, `translateR` uses the `textcat` package (Hornik et al., 2012) to confirm that users have selected the proper source language, a functionality that is very important, given that simple typos in language can could cost the user money when used with large corpora. Finally, `translateR` parallelizes the translation process, thus reducing the runtime necessary to translate an entire corpus. In sum, `translateR` provides a powerful mechanism for `R` users to access cutting edge machine translation tools and in a way that comports with topic modeling software.[11]

C.1 Usage

To use the Google API, users must provide a valid API key, available at `https://developers.google.com/translate/v2/getting_started`. If a Google API key is passed

[10] Discussion of the relevant advantages of each, including access cost, is beyond the scope of this document. Though we note that as of summer 2014, the Microsoft API is that it will accept longer documents.

[11] An alternative, less developed, `R` package for translation is the `translate` package which accesses the Google API only and can only process one document at a time, without any of the additional features discussed above.

to `translate()`, the Google API will be used to translate the text. To use the Microsoft API, users must pass a valid client ID and client "secret value," available at `http://msdn.microsoft.com/en-us/library/hh454950.aspx`. Note that users must pass to `translate()` either a Google key or both a Microsoft client ID and a Microsoft client secret value.

The `translateR` packages installs with a small dataset constructed from the Enron corpus (Klimt and Yang, 2004), a corpus constructed from emails sent during the Enron scandal and in the public domain. The small subset has ten rows and three columns, two of which are metadata (the email subject and date). The text is simply the body of the email. The column names are "email," "date," and "subject." In the following example, we demonstrate different ways in which the software can be used to translate the original text into German. Package help files contain similar documentation.

```
# Load example data. Three columns, the text content ('email')
# and two metadata fields ('date' and 'subject')
data(enron)

# Google, translate column in dataset
google.dataset.out <- translate(dataset = enron, content.field = 'email',
                          google.api.key = my.api.key,
                          source.lang = 'en', target.lang = 'de')

# Google, translate vector
google.vector.out <- translate(content.vec = enron$email,
                          google.api.key = my.api.key,
                          source.lang = 'en', target.lang = 'de')

# Microsoft, translate column in dataset
google.dataset.out <- translate(dataset = enron, content.field = 'email',
                          microsoft.client.id = my.client.id,
                          microsoft.client.secret = my.client.secret,
                          source.lang = 'en', target.lang = 'de')

# Microsoft, translate vector
google.vector.out <- translate(content.vec = enron$email,
                         microsoft.client.id = my.client.id,
                         microsoft.client.secret = my.client.secret,
                         source.lang = 'en',
                         target.lang = 'de')
```

The full set of arguments for `translate()` is as follows, though we refer users to `translateR` package documentation for details.

- **dataset:** A dataframe with a column containing the text to be translated.

- **content.field:** If a dataframe is passed to "dataset", the name of the column containing the text must be passed to "content.field".
- **content.vec:** A character vector of text. This is an alternative to "dataset"/"content.field".
- **google.api.key:** To use the Google API, an API key must be provided. For more information on getting your key, see `https://developers.google.com/translate/v2/getting_started`.
- **microsoft.client.id:** To use the Microsoft API, a client id and a client secret value must be provided. For more information on getting these, see `http://msdn.microsoft.com/en-us/library/hh454950.aspx`. NOTE: you do not need to obtain an access token. translateR will retrieve a token internally.
- **microsoft.client.secret:** To use the Microsoft API, a client id and a client secret value must be provided. For more information on getting these, see `http://msdn.microsoft.com/en-us/library/hh454950.aspx`. The client secret value is a unique identifying string obtained when registering with Microsoft (see the link for more information). NOTE: you do not need to obtain an access token. translateR will retrieve a token internally.
- **source.lang:** The language code that corresponds with the language in which the source text is written. The translators use different language codes, so select accordingly. To see a list of language codes, enter getGoogleLanguages() or getMicrosoftLanguages() for Google or Microsoft, respectively.
- **target.lang:** The language code that corresponds with the language into which the source text is to be translated. The translators use different language codes, so select accordingly. To see a list of language codes, enter getGoogleLanguages() or getMicrosoftLanguages() for Google or Microsoft, respectively.

Note that the `source.lang` and `target.lang` arguments require language codes specific to the Google and Microsoft APIs. To make it easier for users to access these codes, we provide two functions. `getGoogleLanguages()` and `getMicrosoftLanguages()` print the full listing of languages supported by each API and their corresponding code so that users do not have to look this information up online.

D Online Appendix: Core STM Technical Details

We briefly state the data generating process for the STM, referring readers elsewhere for additional details (Roberts et al., 2014, nd, 2013). The generative process for each document (indexed by d) can be summarized as:

1. Draw the document-level attention to each topic from a logistic-normal generalized linear model based on document prevalence covariates X_d.
   $\vec{\theta}_d | X_d\gamma, \Sigma \sim \text{LogisticNormal}(\mu = X_d\gamma, \Sigma)$

2. Form the document-specific distribution over words representing each topic (k) using the baseline word distribution (m), the topic specific deviation κ_k, the content covariate group deviation κ_g and the interaction between the two κ_i.
   $\beta_{d,k} \propto \exp(m + \kappa_k + \kappa_{g_d} + \kappa_{i=(kg_d)})$

3. For each word in the document, $(n \in 1, \ldots, N_d)$:

   - Draw word's topic assignment based on the document-specific distribution over topics.
     $z_{d,n}|\vec{\theta}_d \sim \text{Multinomial}(\vec{\theta})$
   - Conditional on the topic chosen, draw an observed word from that topic.
     $w_{d,n}|z_{d,n}, \beta d, k = z \sim \text{Multinomial}(\beta_{d,k=z})$

The model is completed with regularizing prior distributions for γ, κ and Σ. Both prevalence and content covariates are optional, When content covariates are not included the distribution over β can optionally be optimized directly as in the standard LDA model. Without prevalence or content covariates, the model reduces to a logistic-normal topic model (Blei and Lafferty, 2007) rather than LDA as the Logistic Normal allows the topics to be correlated.

We estimate the model using a variational EM algorithm as implemented in the `stm` package in `R`. All reported covariate effects are estimated in the package as well using a method of composition style approach to dealing with measurement uncertainty in the latent topics (Roberts et al., 2014).

E Online Appendix: Topic Correlation Graph Estimation

Here we introduce and discuss two estimation procedures we provide for producing topic correlation plots both of which are implemented in the `stm` package. The first method is conceptually simpler and involves a simple thresholding procedure on the estimated marginal topic covariance matrix and requires a human specified threshold. The second method draws on recent literature undirected graphical model estimation and can be automatically tuned. For Figure 2 in the paper we use simple thresholding measure.

Simple Thresholding Taking the correlation of the MAP estimates for the topic proportions θ yields the marginal correlation of the mode of the variational distribution. Then we simply set to 0 those edges where the correlation falls below the user threshold. A method of composition approach can be used to integrate over the uncertainty in the topic means by repeatedly drawing from the variational posterior for θ and calculating the correlation. This makes it possible to derive confidence intervals on the topic correlations.

Graph Estimation An alternative strategy is to treat the problem as the recovery of edges in a high-dimensional undirected graphical model. In these settings we assume that observations come from a multivariate normal distribution with a sparse precision matrix. The goal is to infer which elements of the precision matrix are non-zero corresponding to edges in a graph. In an influential piece, Meinshausen and Bühlmann (2006) showed

that using sparse regression methods like the LASSO it is possible to consistently identify edges even in very high dimensional settings. Drawing on this work Blei and Lafferty (2007) use the Meinshausen and Bühlmann (2006) approach to estimate a topic graph in the Correlated Topic Model by running a LASSO regression on the variational means. The variational means are plausibly multivariate normal but are not the true quantity of interest.

We use the recently developed Nonparanormal SKEPTIC procedure which was developed to extend previous results to the non-Gaussian case (Liu et al., 2012; Zhao et al., 2012). Essentially the data are transformed using a Gaussian Copula approach and then tested using a LASSO as in Meinshausen and Bühlmann (2006). Thus we are able to run model selection on the true quantity of interest: the MAP estimates for θ. Model selection for the scale of the L_1 penalty is performed using the rotation information criterion (RIC) which estimates the optimal degree of regularization by random rotations. The authors note that this selection approach has strong empirical performance but is sensitive to under-selection of edges (Zhao et al., 2012). We choose this metric as the default approach to model selection to reflect social scientists' historically greater concern for false positive rates as opposed to false negative rates.

The Nonparanormal SKEPTIC procedure has been shown to have excellent theoretical properties and empirical success in large genomics data sets (Liu et al., 2012). In high-dimensions (a large number of topics in our case) the procedure can be shown to have optimal parametric convergence rates even when the data are truly Gaussian. The procedure also has the advantage of identifying significant negative relationships between topics and effectively visualizing those for the user is a matter of future research.

We note that in models with low numbers of topics the simple procedure and the more complex procedure will often yield identical results. However, the advantage of the Nonparanormal SKEPTIC procedure hat we outline here is that it scales gracefully to models with hundreds or even thousands of topics - specifically the set of cases where some higher level structure like a correlation graph would be the most useful.

F Online Appendix: Additional Information on Empirical Examples

F.1 Fatwah Analysis

The Fatwah texts come as raw Arabic texts, so preprocessing is necessary. We first clean the text of punctuation marks and stray characters, and standardize some variants of Arabic letters that are functionally equivalent. We then remove numbers, reflecting our belief that these are not important for differentiating topics. We also believe that stop words are not important for our goal of uncovering the topics discussed by Muslim clerics so we remove stop words according to a custom list we developed in Arabic. We then remove prefixes and suffixes from remaining words using the "light 10" Arabic stemmer (Larkey et al., 2007) with minor modifications and implemented through the python extensibility available in txtorg. Stemming is appropriate for this application because we expect topics to be primarily related to the core concepts conveyed by each stem,

rather than specific prefixing, infixing, or suffixing. After preprocessing the Arabic text as described, we split the corpus into tokens by dividing the text at spaces in the text and removing any remaining whitespace. From these tokens, we construct a document-term matrix that can be read into stm.

F.2 Snowden Analysis

In this section, we provide additional detail concerning the Snowden analysis, including preprocessing, estimation, and interpretation of additional topics not discussed in Section 4.2, and also alignment across the two translation approaches.

F.3 Preprocessing

In the paper we analyze unique tweets, leaving the analysis of a combined retweet/original tweet corpus to future work which could study the intensity of particular messages. For both the Arabic and Chinese, we removed exact duplicates. However, some retweets are not exact matches, for any number of reasons. To address this issue in the Arabic corpus, we also removed tweets containing the string "RT", which signifies a retweet. For the Chinese corpus, we removed text containing the phrase, zhuanfa weibo, which is used to identify retweets, translating "forwarded Weibo."

To preprocess the Chinese text for full text translation, after translating the text, we stemmed the corpus and removed stopwords and infrequent words. For the document-term matrix translation, we first segmented the chinese corpus using the Stanford Word Segmenter (Chang et al., 2008; Tseng et al., 2005), though we also could have accomplished this with `txtorg`. Then, with both the Chinese and Arabic corpora, we converted the remaining terms to lowercase, stemmed them, removed stopwords, and removed all terms less than two characters in length. Following this procedure, we created a document-term matrix from the segmented corpus, then translated the vocabulary and combined any columns (terms) that translated into the same English term. We removed infrequent terms and then conducted the analysis on the remaining.

F.4 Alignment of Asylum, Human Rights, and Attack topics

To examine the alignment between topics estimated with full text translation versus document-term matrix translation, we marginalize over the betas, convert the vocabulary to lower case, and merge the betas by vocabulary. We then calculate the correlation across the two translation methods and generate the correlation plot displayed in Figure2.

The "Human Rights" topic in full text translation is that with the frequent words "american," "countri," and "peopl," which matches to the "Human Rights" topic in the DTM translation with frequent words "american," "country," and "traitor." This match can be seen in row 3, column 5.

The "Attack" topic in full text translation is that with the frequent words "china," "newspap," and "snowden," which matches to the "Attack" topic in the DTM translation with frequent words "china," "american," and "chinese." This match can be seen in row 15, column 1.

The "Asylum" topic in full text translation is that with the frequent words "asylum," "ecuador," and "snowden," which matches to the "Asylum" topic in the DTM translation

with frequent words "ecuador," "shelter," and "political." This match can be seen in row 12, column 10.

As evident in the figure, each pair matches well with its partner but the topics of interest are also sometimes correlated with a third topic. Thus, they appear to be identifying a common underlying topic, though we caution readers from drawing too strong a conclusion, as they are not (and cannot be expected to be) perfectly correlated.

F.4.1 Additional topics

Figure 3 displays the topics and the expected topic proportions for the full text translation, while Figure 4 does the same for the DTM translation. In Figure 3, the "Human Rights," "Attack," and "Asylum" topics (discussed at length in Section 4.2) are the 7^{th}, 12^{th}, and 5^{th}, respectively (counting down from the top). In the Figure 4, they are the 1^{st}, 8^{th}, and 5^{th}, respectively.

For the sake of brevity, we discussed only those topics in the body of the paper. However, other topics are potentially interesting. For instance, in the text translation, we observe a topic about signing a petition to stand with Snowden and stop the fighting. The "Petition" topic is associated with terms like "snowden", "spi", "share", "edward", "accus", "stop", and "pheoniz". One related tweet on Weibo reads (after translation), "Stand with Edward Snowden # #Snowden! BarackObama - Stopped spying #NSA. Signed the petition and share." It has an expected topic prevalence of 6.22%.

G Online Appendix: Benchmarking common pre-processing tools

As the total volume of text increases, the ability to efficiently manage text at scale becomes important. We argue here that once researchers start to work with larger volumes of text, an index-based management system like that provided by `txtorg` quickly becomes essential. An index is a data structure that allows the user to search the full corpus without without scanning every document, allowing for fast search operations even with millions of documents. The Lucene-powered index is essentially represented as a file containing a list of all the terms in the corpus, along with a link to the specific documents in which that term appears. Thus, instead of scanning every document in search of a particular term, `txtorg` scans the list of all the unique terms in the index, locates the match, and returns the documents linked to that term. In addition to searches through the corpora, `txtorg` also permits fast searches of metadata and joint searches of both metadata and text. `txtorg` can handle complex queries, including regular expressions, booleans, and a host of Lucene-supported options.

Once `txtorg` reads in a corpus, a new Lucene index is created and the original corpus may be discarded or stored elsewhere. This new index is considerably smaller than the original data permitting local storage of what might otherwise be an unwieldy corpus.[12] From the Lucene index, a user can export a DTM even if the original documents are not available.

[12]For example, we index an Arabic corpus of size 195.3 MB, which yields an index of size 78.1 MB. The relationship between corpus size and index size is approximately logarithmic rather than linear.

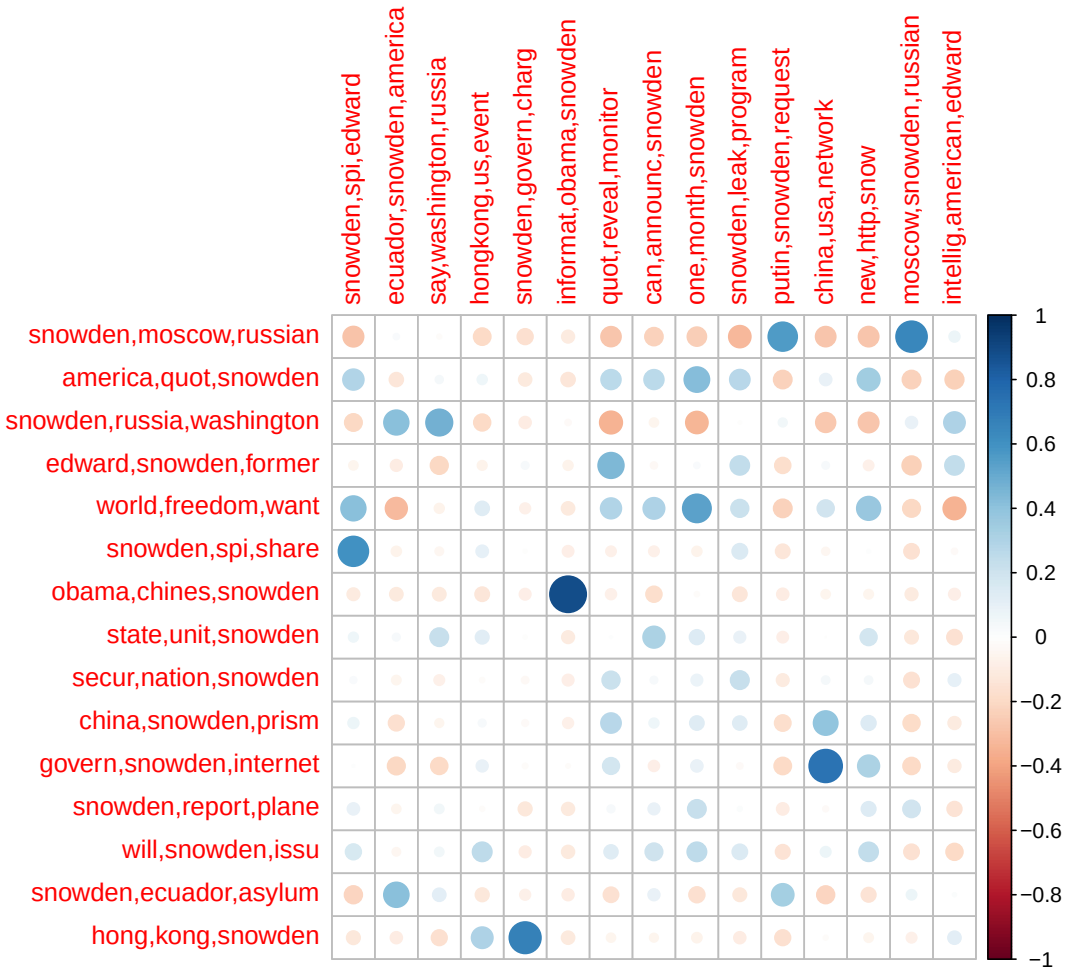


Figure 2: Correlation between topics in both translation approaches. The vertical axis labels topics from full text translation and the horizontal access labels those from document-term matrix translation.

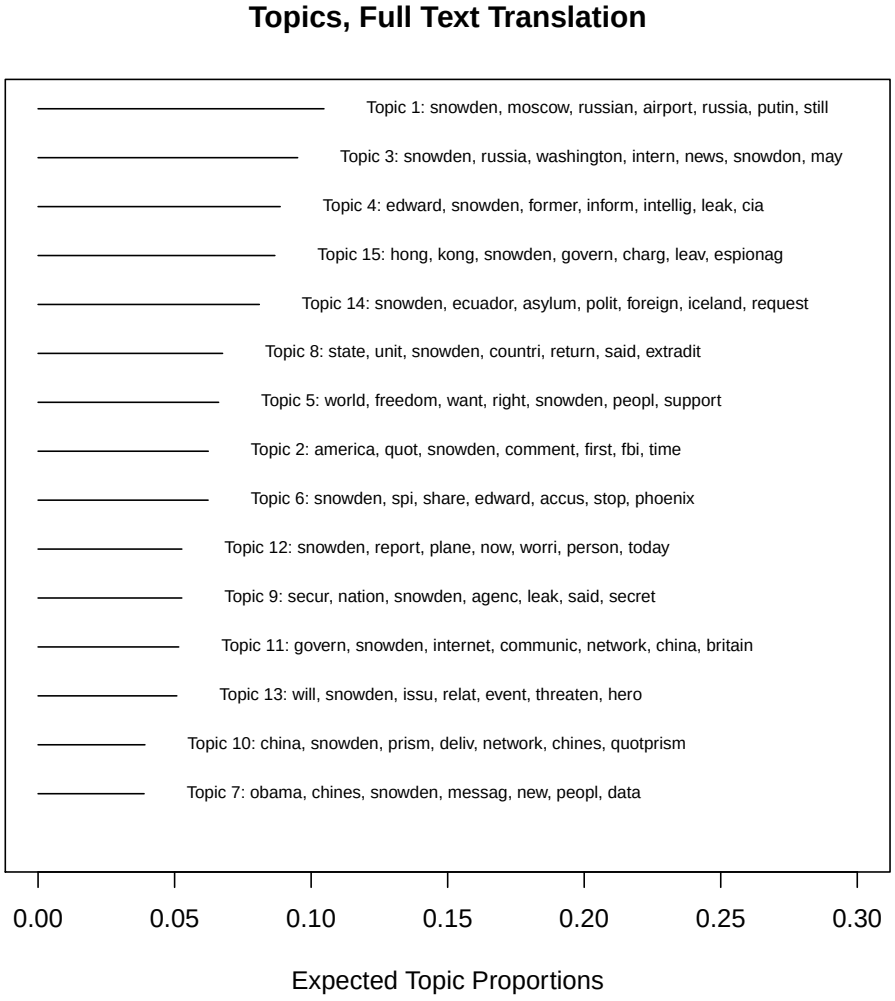


Figure 3: All Topics for Snowden Full Text Translation Model.

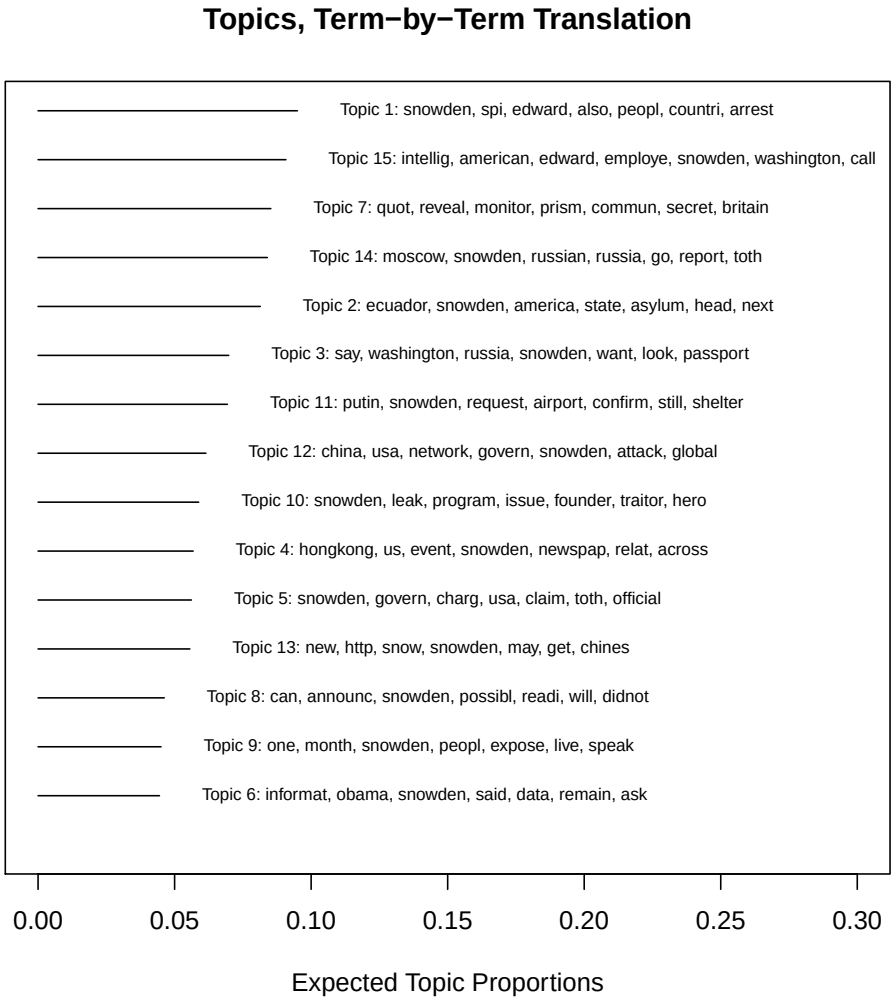


Figure 4: All Topcis for Snowden DTM Translation Model.

Software	txtorg		TM	
	Index	DTM	Corpus Object	DTM
January, 1994 (one month; 6,599 documents)	14.83s	53.23s	34.606s	34.629s
January - December, 1994 (twelve months; 75,303 documents)	174.83s	867.34s	152.566s	1232.507s
January 1994 - December 1998 (five years; 389343 documents)	853.65s	4023.47s	922.255s	FAILED
January 1994 - December 2003 (ten years; 802353 documents)	1643.83s	7235.98s	FAILED	FAILED
Number of languages stemmed	>50		14	
Number of languages stop word removal	>50		15	
Native segmentation support	Chinese, Japanese, Korean		✗	
Index based fast searching	✓		✗	
Outputs associated meta-data	✓		✗	
Multi-corpora management	✓		✗	

Table 3: Results from benchmarking `txtorg` against `TM` on three ranges of New York Times data. All times in seconds. Document length will have an effect on run times. FAILED indicates that R either froze or threw a memory allocation error.

A direct consequence of the index based system is speed in producing the DTM. To demonstrate the speed advantages of an index based system, we benchmark `txtorg` against the commonly recommended `R` package, `TM` (Feinerer et al., 2008).[13] `TM` provides a framework for preprocessing, in which corpora are read into `R`, after which they may be preprocessed and then converted into a DTM object. Though not perfectly analogous, we compare the time necessary to create a corpus object in `TM` to the time necessary to create an index in `txtorg`, and compare the time necessary to create a DTM object in `TM` to that necessary to write a DTM in `txtorg`. We compare the runtime for `TM` to that of `txtorg` on several differently sized corpora from the New York Times. The results from this comparison are shown in Table 3, which clearly demonstrates that while the construction of the index can be nontrivial, it provides a flexible, fast framework for the creation of document-term matrices. Moreover, because users only need to create an index once for any given corpus, moving the computational burden to the indexing stage is prudent, as it makes all subsequent operations faster.[14]

[13] `TM` is a superb package, offering a range of functions useful for text mining. But because it is not an index based system, it suffers in terms of speed and ability to handle large corpora.

[14] `txtorg` outputs several different file formats - primarily, a sparse matrix format and a flat CSV, and hence the corpus can be exported in a form easily read into any major text processing software such as the `Topicmodels` package in `R`.

References

Beauchamp, N. Text-based scaling of legislatures: A comparison of methods with applications to the us senate and uk house of commons.

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

Blei, D. M. and J. D. Lafferty (2007). A correlated topic model of science. *The Annals of Applied Statistics 1*(1), 17–35.

Chang, J. (2012). *lda: Collapsed Gibbs sampling methods for topic models.* R package version 1.3.2.

Chang, P.-C., M. Galley, and C. D. Manning (2008). Optimizing chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pp. 224–232. Association for Computational Linguistics.

Converse, P. E., W. E. Miller, J. G. Rusk, and A. C. Wolfe (1969). Continuity and change in american politics: Parties and issues in the 1968 election. *The American Political Science Review 63*(4), 1083–1105.

Elff, M. (2013). A dynamic state-space model of coded political texts. *Political Analysis 21*(2), 217–232.

Feinerer, I., K. Hornik, and D. Meyer (2008, March). Text mining infrastructure in r. *Journal of Statistical Software 25*(5), 1–54.

Gentzkow, M. and J. M. Shapiro (2010). What drives media slant? evidence from us daily newspapers. *Econometrica 78*(1), 35–71.

Grimmer, J. (2010). A bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases. *Political Analysis 18*(1), 1.

Grimmer, J. and B. M. Stewart (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis 21*(3), 267–297.

Grün, B. and K. Hornik (2011). topicmodels: An R package for fitting topic models. *Journal of Statistical Software 40*(13), 1–30.

Hillard, D., S. Purpura, and J. Wilkerson (2008). Computer-assisted topic classification for mixed-methods social science research. *Journal of Information Technology & Politics 4*(4), 31–46.

Hopkins, D., G. King, M. Knowles, and S. Melendez (2010). Readme: Software for automated content analysis. *Institute for Quantitative Social Science*.

Hopkins, D. J. and G. King (2010). A method of automated nonparametric content analysis for social science. *American Journal of Political Science 54*(1), 229–247.

Hornik, K., J. Rauch, C. Buchta, and I. Feinerer (2012). textcat: N-gram based text categorization. *R package version 0.1-1*.

Jamal, A., R. O. Keohane, D. Romney, and D. Tingley (n.d.). Anti-americanism or anti-interventionism? evidence from the arabic twitter universe. *Perspectives on Politics forthcoming*.

Jensen, J., E. Kaplan, S. Naidu, and L. Wilse-Samson (2012). Political polarization and the dynamics of political language: Evidence from 130 years of partisan speech. *Brookings Papers on Economic Activity 2012*(2), 1–81.

Johnston, A. I. and D. Stockmann (2007). Chinese attitudes toward the united states and americans. *Anti-Americanisms in world politics*, 157–95.

Jurka, T. P., L. Collingwood, A. E. Boydstun, E. Grossman, and W. van Atteveldt (2013). *RTextTools: Automatic Text Classification via Supervised Learning*. R package version 1.4.1.

Kesselman, M. (1966). French local politics: A statistical examination of grass roots consensus. *The American Political Science Review 60*(4), 963–973.

King, G., J. Pan, and M. E. Roberts (2013). How censorship in china allows government criticism but silences collective expression. *American Political Science Review 107*, 1–18.

Klimt, B. and Y. Yang (2004). The enron corpus: A new dataset for email classification research. In *Machine learning: ECML 2004*, pp. 217–226. Springer.

Krippendorff, K. (2012). *Content analysis: An introduction to its methodology*. Sage.

Larkey, L. S., L. Ballesteros, and M. E. Connell (2007). Light stemming for arabic information retrieval. In *Arabic computational morphology*, pp. 221–243. Springer.

Laver, M., K. Benoit, and J. Garry (2003). Extracting policy positions from political texts using words as data. *American Political Science Review 97*(02), 311–331.

Liu, H., F. Han, M. Yuan, J. Lafferty, and L. Wasserman (2012). High-dimensional semiparametric gaussian copula graphical models. *The Annals of Statistics 40*(4), 2293–2326.

Lowe, W. (2008). Understanding wordscores. *Political Analysis 16*(4), 356–371.

Lowe, W. (2013). *Austin: Do things with words*.

Lowe, W. and K. Benoit (2013). Validating estimates of latent traits from textual data using human judgment as a benchmark. *Political Analysis 21*(3), 298–313.

Meinshausen, N. and P. Bühlmann (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics 34*(3), 1436–1462.

Monroe, B. L., M. P. Colaresi, and K. M. Quinn (2008). Fightin'words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis 16*(4), 372–403.

Pennebaker, J., M. Francis, and R. Booth (2001). *Linguistic Inquiry and Word Count: LIWC 2001*. Mahway, NJ: Erlbaum Publishers.

Quinn, K., B. Monroe, M. Colaresi, M. Crespin, and D. Radev (2010). How to analyze political attention with minimal assumptions and costs. *American Journal of Political Science 54*(1), 209–228.

Roberts, M. E., B. M. Stewart, and E. Airoldi (n.d.). A topic model for experimentation in the social sciences. *working*.

Roberts, M. E., B. M. Stewart, D. Tingley, and E. M. Airoldi (2013). The structural topic model and applied social science. *Advances in Neural Information Processing Systems Workshop on Topic Models: Computation, Application, and Evaluation*.

Roberts, M. E., B. M. Stewart, D. Tingley, C. Lucas, J. Leder-Luis, S. Gadarian, B. Albertson, and D. Rand (2014). Structural topic models for open-ended survey responses. *American Journal of Political Science 58*(4), 1064 – 1082.

Slapin, J. B. and S.-O. Proksch (2008). A scaling model for estimating time-series party positions from texts. *American Journal of Political Science 52*(3), 705–722.

Spirling, A. (2011). Us treaty making with american indians: Institutional change and relative power, 1784–1911. *American Journal of Political Science*.

Spirling, A. (2012). Comment on 'political polarization and the dynamics of political language: Evidence from 130 years of partisan speech'. *Brookings Papers on Economic Activity 2012*(2), 1–81.

Stockmann, D. (2011). Race to the bottom: media marketization and increasing negativity toward the united states in china. *Political Communication 28*(3), 268–290.

Stone, P., D. Dunphy, M. Smith, and D. Ogilvie (1966). *The General Inquirer: A Computer Approach to Content Analysis*. The MIT Press.

Taddy, M. (2013). Multinomial inverse regression for text analysis. *Journal of the American Statistical Association 108*(503), 755–770.

Tseng, H., P. Chang, G. Andrew, D. Jurafsky, and C. Manning (2005). A conditional random field word segmenter for sighan bakeoff 2005. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, Volume 171. Jeju Island, Korea.

Wallach, H. M. (2006). Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pp. 977–984. ACM.

Zhao, T., H. Liu, K. Roeder, J. Lafferty, and L. Wasserman (2012). The huge package for high-dimensional undirected graph estimation in r. *The Journal of Machine Learning Research 98888*, 1059–1062.