Computer assisted text analysis for comparative politics

This draft: December 31, 2013

Abstract

Comparative politics scholars are well poised to take advantage of recent advances in research designs and research tools for the systematic analysis of textual data. This paper provides the first focused discussion of these advances for scholars of comparative politics, though many arguments are applicable across political science sub-fields. With the explosion of textual data in countries around the world, it is important for comparativists to stay at the cutting edge. We situate recent and existing tools within a broader framework of methods to process, manage, and analyze textual data. While we review a variety of analysis tools of interest, we particularly focus on methods that take into account information about when and who generated a particular piece of text. We also engage with more pragmatic considerations about the ability to process large volumes of text that come in multiple languages. All of our discussions are illustrated with existing, and several new, software implementations.

1 Introduction

In this paper we focus on new tools for comparativists to utilize *textual* data that research designs in comparative politics generate. 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 its own unique challenges, including the incorporation of metadata, dealing with text data on a large scale, and analyzing text in multiple languages.

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). In this paper we lay out different types of text analysis that have been used by comparativists. We discuss and provide examples for two broad categories of methods: supervised and unsupervised learning. While a subset of the methods we discuss have been applied within comparative politics, we highlight unsupervised topic modeling as a relatively underutilized approach. We provide a more extensive discussion of a particular new method within this class, the Structural Topic Model (STM) [Blinded for review], along with several original examples. We argue below that the STM should be an important part of the text analysis tool kit for comparativists.

The Structural Topic Model is an unsupervised method for uncovering thematic structure within a corpus of documents. Topic models discover distributions over words, or "topics", which can serve as a semantically meaningful low-dimensional summary of document contents. Crucially, the topic model does not require the analyst to pre-specify the topics; rather, the contents of the corpus are discovered from the documents themselves based on the patterns of frequently co-occurring words. The ability to quickly summarize documents without needing to specify categories manually makes topic models an important part of the text analysis tool kit. But prior to the STM, topic models did not easily include relationships between topics and "metadata" associated with the text, such as when the text was written, where it was written, who wrote it, and characteristics of the author. The STM provides a flexible way to incorporate this information into the

analysis using document-level covariates. In turn, it allows comparativists to understand relationships between metadata and topics in their text corpus. Throughout our discussion of supervised, scaling, and unsupervised methods we supply ample examples to help readers understand the differences between existing approaches and identify methods that will be helpful for their own projects. We devote particular attention to two example applications of the Structural Topic Model which exemplify the underutilized potential of unsupervised methods.

A second contribution of the paper is a discussion of principles for text management. We underscore the importance of management tools that can support multiple languages and scale to large datasets. We introduce one such tool, `txtorg`, that allows users manage large volumes of textual data. `txtorg` was designed with comparativists directly in mind because its processing tools support a broad variety of languages obviating the need for individual, application-specific solutions. `txtorg` uses the Apache `Lucene` engine as a back-end allowing it to process millions of documents which other tools are unable to handle. Scalability is particularly salient for scholars who want to study newspapers, social media or microblogging (e.g. Twitter, China's Sina Weibo). `txtorg` provides a clean and efficient way to manage textual data alongside "metadata" about the text itself.

The structure of the paper is as follows. Section 2 discusses *research questions* in comparative politics that have benefited from text analysis tools and a multi-language view of *text processing*. Section 3 presents a heuristic review of text *analysis tools*. This sets up the discussion in Section 4 which introduces the STM model in detail, and provides two example analyses using Islamic fatwas and newspaper reporting on Tibet and SARS.

2 Text and Language Basics

2.1 Research Questions and Data Analysis

Automated content analysis and comparative politics are an ideal pair. Countries around the world are producing textual data at unprecedented rates. Traditional government statistics are often missing, mis-measured, 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 have 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 this 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 England 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.

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

this text data to answer comparative questions about political systems (Budge, 2001). Early versions used human coding, but more recently the CMP and related projects have been assisted by computer techniques. Catalinac (2013) uses thousands of Japanese election manifestos from 1986 to 2009 to determine how electoral strategies shifted after Japan's electoral reform in 1994. Nielsen (2012) 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) study the focus of censorship in social media in China, Jamal et al. (2013) study anti-Americanism in Arabic-language Twitter posts, and Barberá (2012) uses Twitter posts to scale citizen liberal-conservative ideal points across the US 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 Multi-language 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 touted as language agnostic, however, more seldomly discussed are the language specific tools for pre-processing. While each language has its own set of tools, there are a set of steps common to reprocessing in almost all languages. We discuss three challenges that must be overcome in all languages: creating machine-readable text, pre-processing for dimensionality reduc-

tion, and handling of large corpora. We then point out language-sepcific variations that comparativist studying particular countries should consider.

2.2.1 Mechanics to Make Text Machine Readable

Particularly for comparativists who extract text from a variety of different sources, the text they gather originally 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 technologies may be required.[1] Even if they have been digitized, the text may be from different encodings and therefore not immediately usable as one corpus.

The encoding of text is the way in which the computer translates individual, unique characters into bytes. Each language can have multiple encodings[2] and different computers and different software will default to recognizing 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.[3] 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.

[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, with clearly-written text scholars 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, error 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.

[2]For example Chinese has several dozen encodings, the largest of which are Guobiao (GB), which has a two or four byte encoding, Big5 which has a one or two byte encoding, and ISO-2022, which has a seven byte encoding.

[3]Most programming languages have packages to transfer between encodings. For example, to convert encodings we use Python's package *chardet*.

2.2.2 Pre-processing 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 inherent dimensionality of natural language text to the length of dictionary with one entry per unique word in the corpus. Unfortunately, even these dictionaries can be too large to be practical, ranging from thousands to millions of unique words. Bounding the size of the vocabulary by combining similar terms and removing irrelevant words is crucial for automated content analysis methods to work well in practice. We describe the most common tools 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.[4]

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 setting this involves removing frequently occurring function words such as "and" and "the."[5] Most languages have lists of common "stop words" that can be provided to programs such as `txtorg` to remove stopwords.

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

[4]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.

[5]In other settings, such as the analysis of style or authorship detection, function words may be the sole quantity of interest.

order to convey a tense or 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 Schütze, 2008).[6]

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 Schütze, 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[7] 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 "Kirch," or church, can be appended to "rat," forming "Kirchrat" who is a member of the church council, or

[6]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.

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

"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. The analyst might wish to compound these into "nationalsecurity" and "socialsecurity", but 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 pre-processing has been completed, for many automated content techniques (including those detailed in this paper), the remaining words are used to construct a document-term matrix (DTM). A document-term matrix 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. The document-term matrix is the primary input to most automated text analysis methods.

2.3 Management of Textual Data

All of the previous steps are not trivial from a workflow perspective, especially for comparativists working in a variety of languages with large text corpora. In this section, we discuss existing methods to deal with these management issues and introduce `txtorg`, an open source tool for text preprocessing and management with particular value for research in comparative politics. `txtorg` breaks a common bottleneck between unprocessed corpora and the subsequent analysis.

2.3.1 Multi-language pre-processing

At the core of `txtorg` is Apache Lucene (Cutting et al., 2013), a high-performance text search engine library. By drawing on the active open source Lucene community, `txtorg` is able to provide support for a diverse set of languages. `txtorg` currently includes support for Arabic, Bulgarian, Portuguese (separate tools for Brazil and Portugal), Catalan, Chinese, Japanese, Korean, Czech, Danish, German, Greek, English, Spanish, Basque, Persian, Finnish, French, Irish, Galician, Hindi, Hungarian, Armenian, Indonesian, Italian, Latvian, Dutch, Norwegian, Romanian, Russian, Swedish, Thai, and Turkish. Moreover `txtorg` will be able to incorporate new developments from the Lucene community as they become available.

As discussed in Section 2.2, each language poses unique challenges to pre-processing. `txtorg` leverages the dedicated language-specific preprocessing utilities (stemming, segmentation, etc) that have been created by the open source Lucene community. Thus, in addition to the canonical Porter stemmer used in most English texts (Porter, 1980), we provide, for example, a German stemmer and an Arabic stemmer, each optimized for their respective language. These "analyzers" are essentially a composite of preprocessing decisions. While there are numerous "out of the box" analyzers that any user can apply to his or her text with very little overhead, users can also write their own custom analyzers, incorporate them into `txtorg`, and distribute them to the larger open source Lucene community. `txtorg` also provides support for manual and automated spelling correction in

English,[8] the substitution of prespecified strings with corrected strings, and the indexing of multi-word phrases.

2.4 Index Based Management

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.[9] From the Lucene index, a user can export a DTM even if the original documents are not available.

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, Hornik and Meyer, 2008).[10]

[8] The algorithm for automated spelling correction is a simple probabilistic one that works on any language where terms are strings of characters. Because it relies on a training set in the relevant language, users can extend the algorithm to other languages by uploading a training corpus in the appropriate language. The `txtorg` documentation details this process.

[9] 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.

[10] `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.

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 1: 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. Processing done on an ASUS 1015E netbook. FAILED indicates that R either froze or threw a memory allocation error.

`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 1, 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.[11]

Most data in comparative politics come with an implicit structure. `txtorg` supports this metadata. Structural metadata is perhaps best understand as information about the relationships between documents in the full corpus. Examples include data about the document's author, its place of origin, or any other characteristic that might relate to its generation. Traditional approaches, like Latent Dirichlet Allocation (Blei, Ng and Jordan,

[11] `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`.

2003), are unable to deal with this structure in a principled way (which we discuss in more detail below). We give examples in Section 4 of how to incorporate this information into analysis. `txtorg` manages the metadata and exports a correctly formatted metadata file that can be read into the `STM` package in `R`.

3 Types of Computer Assisted Text Analysis

The previous section demonstrated how to represent a corpus in a document-term matrix where each row represents a document and contains a vector of counts one for each word type. While these preprocessing steps dramatically reduce the dimensionality of the text, our theoretical questions can rarely be answered by counting how many times a particular word is used. Thus we turn to more nuanced forms of dimensionality reduction for inferring corpus-wide patterns and producing interpretable summaries of our documents. Our choice of method will be dictated by the type of summary we are hoping to extract.

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, we do not specify the conceptual structure of the texts beforehand, instead allowing the model to find a low-dimensional summary that best explains observed word counts (under some set of assumptions). Consequently human effort is focused on interpretation of the results after the model is run.

In this section, 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. 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".[12] 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. In Section 4 we develop a more detailed case study of a new method, the Structural Topic Model, and provide two applications within comparative politics.

3.1 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)[13] 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

[12]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.

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

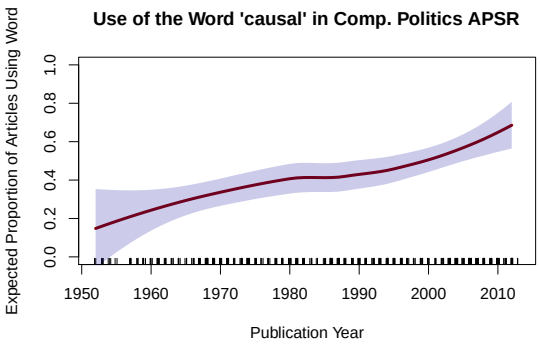

Figure 1: Illustration of Simple Keyword trends. This shows the rise in the proportion of articles in *APSR* using the word 'causal.' This is a suggestive trend which accords with our understanding of the literature but requires additional supervision from the researcher to determine how much we can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

methods over the last 6 decades. While informative, it is important to emphasize that 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, Francis and Booth, 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.[14]

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 words are more predictive of co-authorship.[15] We give the top 10 words for each side in Table 2. 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 2: 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

[14]Monroe, Colaresi and Quinn (2008) provide a comparison of various weighting methods of words for the purposes of feature selection.

[15]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.

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.

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, Benoit and Garry, 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 (N.d.); 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.[16] 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 man-

[16]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, Purpura and Wilkerson, 2008).

ually inspecting new documents and comparing the classifier's coding to manual coding. 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. However, as in the example applications we describe below, the population level statistics are often the quantity of interest in social science settings.

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, Pan and Roberts (2013) provide one recent example of using `ReadMe` in comparative politics. King, Pan and Roberts (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. (2013) 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. (2013) 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 classification errors within the algorithm can be straightforwardly assessed by comparing the algorithm's decisions to the human-created codebook.[17] For keyword methods using preconstructed 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).[18]

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

[18]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.

3.2 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) .

Discovering Corpus Structure Where the input to supervised methods is quite straightforward (a set of hand-coded examples), the input for unsupervised methods is more subtle. In a very general sense, unsupervised methods have the user specify the type of organizing structure that will be most useful. This often takes one of two forms: the specification of what it means for documents to be "similar" or a probabilistic data generating process, or story, about how the data might have been generated. The two approaches are tightly coupled; probabilistic models imply a notion of similarity. We will generally favor the second framework and give a heuristic example of it next.

To help understand how unsupervised learning works, let's take a very simple example shown in Figure 2 of 100 points in a two dimensional case. Let's say that we wanted to create a method to divide these points into two groups. We can write down a *generative* model for each point as follows:

- Randomly choose a group (Group 1 or Group 2)
- Draw a location for the individual point from a distribution centered on the group mean.

Based on this generative story, we can infer where the group means are located and to which group each point belongs. Note we do not have to explicitly define where each

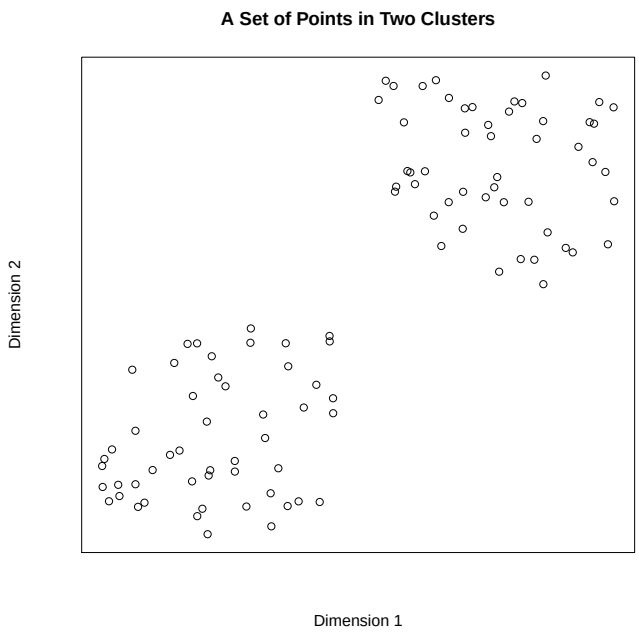

Figure 2: 100 points in an arbitrary two dimensional space.

group is located, we need only specify a probabilistic model where points will tend to be close to their group centers which can in turn be estimated.

Thus a sample model for the data points, denoted by y, might introduce a group indicator we will call z. Then, we might say:

- For each point indexed by i
  - Draw $z_i \sim$Bernoulli(π)
  - Draw $y_i \sim$Normal(μ_z, Σ_z)

Statistical inference would then focus on learning the group membership for each latent group indicator z_i as well as the parameters that define the clusters themselves $(\pi, \mu_1, \mu_2, \Sigma_1, \Sigma_2)$.

Of course this is an extremely simplistic example but we will return to the idea of telling a story about the data generating process to set up more complex examples later.

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. [19]

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.[20]

For our APSR corpus we run an LDA model with six topics and in Table 3 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 3: 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)

[19] 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).

[20] To help distinguish the two types of models we use the term "cluster" when discussing single membership models and "topic" when describing "mixed membership models."

and it also draws from the 5th topic on behavior and the 6th topic on parties and voting.

We often have rich sources of information about our texts beyond simply the words they contain. We refer to this additional data about our texts as "metadata." Metadata can be at the document-level (e.g. the author) or even at the level of individual word tokens (e.g. is the word part of a section heading). Metadata often serves the role of providing the corpus or document with an observable structure that is of intrinsic interest to researchers. Yet most traditional methods provide no clear approach to including this information into the analysis. Recently political scientists have developed new methods for exploring specific types of metadata such as the document's author (Grimmer, 2010) and date Quinn et al. (2010).[21]

There have been comparatively few applications of unsupervised topic modeling within comparative politics. Thus in the next section we highlight a new approach to the topic modeling problem which allows the inclusion of arbitrary metadata and therefore is well-suited to comparative politics. We then demonstrate the method on two applications within comparative politics.

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). In Section 4 we introduce a new method and software tools for estimation and validation of unsupervised topic models using covariates. Our package suggests documents

[21]The appendix for Grimmer (2010) actually includes a variation of the model for including arbitrary author covariates but the primary focus of the model is on the author.

about each topic for you to read and `txtorg` can quickly retrieve documents containing any word or phrase even from extremely large corpora. These tools reinforce the role of unsupervised models as *computer-assisted* reading (Grimmer and King, 2011).

3.3 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.

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.[22] Thus far we have given a brief overview of existing applications of text analysis in comparative politics (Section 2.1), approaches to text processing in other languages (Section 2.2), and how to analyze those texts in an automated way using supervised and unsupervised methods (Section 3). Next in Section 4 we provide examples where we demonstrate how

[22]For approaches to validation see Grimmer and Stewart (2013); Quinn et al. (2010); Lowe and Benoit (2013).

to leverage metadata information, which we see as crucial for comparativists, to answer specific research questions. Specifically, we discuss how to use metadata information in an unsupervised framework, as well as demonstrate a number of interesting quantities of interest that are available by analyzing several data sets.

4 The Structural Topic Models for Comparativists

In this section we describe how the recently introduced Structural Topic Model (STM) [Blinded for review]is an especially useful tool for comparative politics scholars. The STM is a mixed-membership topic model similar to the Latent Dirichlet Allocation (LDA) model discussed above. This means that the model uncovers latent topics in a collection of texts, allowing for each text to express multiple topics. Most previous models make the statistical assumption that documents are *exchangeable*, which means, for example, that two tweets from the same author are no more likely to share a topic than any other two tweets. By contrast, comparativists generally have rich information about the different context in which their documents are written. Using the STM, scholars can incorporate information about each document directly into the topic model.

For example, consider a set of English language texts, such as newspaper reports or Tweets, across a broad range of countries. One might posit that some of these countries are more likely to express some topics compared to others. Using the STM, such a hypothesis can be directly tested using conventional hypothesis testing procedures. Thus the STM allows scholars to straightforwardly test hypotheses about spatial variation.

Comparativists may also be interested in when a document was published or written. For example, consider a hypothesis that suggests that a certain topic is more likely to be mentioned later on in a time series, whereas another topic is more likely early on. The STM straightforwardly incorporates this type of information into the model. Furthermore, with any continuous covariate like time, the STM can fit arbitrarily nonlinear forms.

Hence the STM allows for topic modeling that can test hypotheses about both spatial and temporal variation. This is a distinct advantage for comparativists over previous methods that either assumed that all texts were exchangeable (which is an uninteresting

case for those wishing to compare) or only allowed for a particular type of variation.[23] Before moving on and discussing examples of the STM in action, we first briefly review some technical details.

The Model STM shares the setup of LDA in that each word is assigned to a single topic and documents are represented by mixtures of topics. It differs in allowing document-level metadata 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 in 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). Covariates in topical content allow for the rates of word use to differ by covariate (e.g. women are more likely to use a particular word when talking about topic 1 than men).

The incorporation of metadata permits the pooling of information between documents with similar covariate profiles but it doesn't "bake it in." This allows the model to help researchers find interesting and meaningful covariation in topics, without forcing the metadata to be influential on the estimated topic.[24] Thus the STM is firmly situated in the domain of unsupervised learning.[25] For intuition, if we fit the STM with a randomly sampled covariate, the effect would go to zero (due to the regularizing priors) and the result would be a standard topic model.[26]

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 GLM based on document covariates X_d.

[23]Two special cases have been developed 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 and extends these setups for the mixed-membership case.

[24]This is in contrast to models such as Supervised LDA (Blei and McAuliffe, 2010) where the topics are forced to explain the covariates.

[25]One might be tempted to term it semi-supervised learning, but this references a rather different setup where some data is pre-labeled or partially labeled with topics. In the LDA context Labeled LDA (Ramage et al., 2009) and partially Labeled LDA (Ramage, Manning and Dumais, 2011) span this continuum.

[26]Specifically it would be a correlated topic model (Blei and Lafferty, 2007) rather than LDA as the Logistic Normal allows the topics to be correlated.

$\vec{\theta}_d | X_d\gamma, \Sigma \sim \text{LogisticNormal}(\mu = X_d\gamma, \Sigma)$

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 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. These help enhance interpretation and prevent overfitting. Figure 3 provides a graphical representation of the model.

Topic and Document Level Quantities of Interest The model produces two central quantities of interest at the topic and document levels: (1) a distribution over words for each topic (e.g. the word "institutions" is the most probable under topic one of our APSR corpus (Table 3) and (2) a document-level topic membership vector (e.g. for a particular document 40% of its words come from topic 1, 30% from topic 2 and 30% from topic 3). The first quantity helps us label the topics. The second quantity can be linked to the effect of a covariate on propensity to discuss a given topic. Thus the end result can be standard regression tables (or figures) where a covariate has some relationship to a topic-membership variable.

We can then incorporate our uncertainty in estimating the topic membership vector into our standard regression model using the method of composition. This approach is used by Treier and Jackman (2008) to account for uncertainty in the estimation of a democracy latent variable. In our context, when calculating the effect of a covariate, such

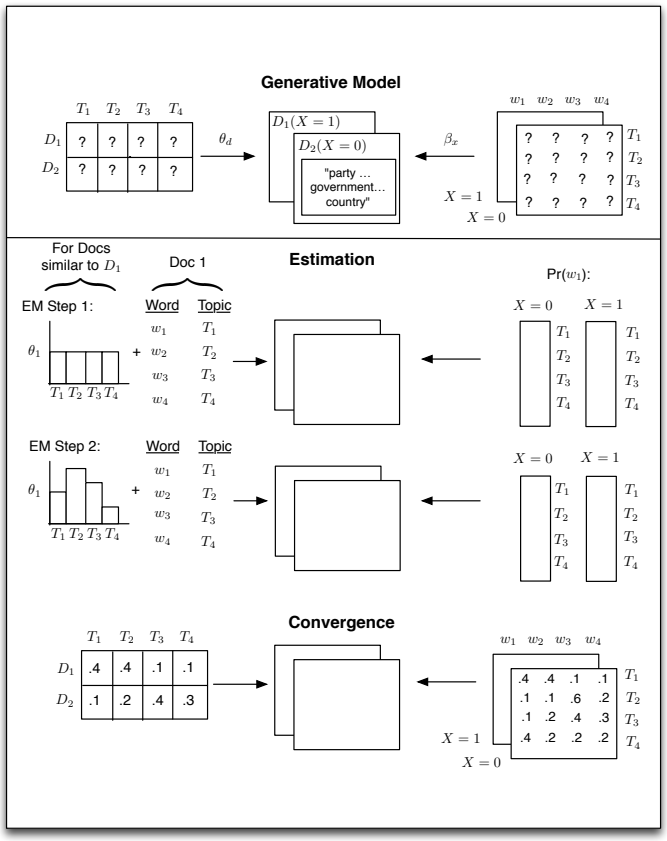

Figure 3: Heuristic description of generative process and estimation of STM.

as an experimental treatment condition, on propensity to discuss a particular topic, we can incorporate our uncertainty in assigning each document's topic membership vector using a simple monte carlo procedure where we sample from the variational posterior of the latent variable.[27]

We estimate the model using a fast deterministic approach (semi-collapsed variational EM). Mixed membership models do not in general have convex likelihoods and consequently results can be sensitive to different initializations. Furthermore, human evaluation experiments in the computer science literature (Chang et al., 2009) has shown that models which provide the best statistical fit can be less interpretable. Consequently

[27]This uncertainty captures the uncertainty in the document's topic composition conditional on the global distributions over words. This implies that the procedure captures the uncertainty due to lower word counts with our certainty about topic composition increasing with document length. This can be quite useful when document's have a variety of different lengths. However this should not be confused with an uncertainty measure that captures epistemic uncertainty over the nature of the topics themselves.

we emphasize a model selection procedure based on interpretability of the model (Grimmer and Stewart, 2013). Our model selection framework focuses on two aspects of the model: *coherence* and *exclusivity* of the topics. In a general sense, coherence favors topics where high probability words often co-occur within the same document and exclusivity favors sets of topics where topics share relatively few high probability words in common.[28]

Corpus Level Quantities of Interest The STM model also calculates quantities of interest at the corpus level, enabling comparativists to take a bird's eye view of their corpus. As described above, the model estimates a topic proportion vector for each document. These proportions can be averaged across documents to get the expected topic proportion for each topic, over all documents in the corpus. The software we provide allows the researcher to plot the topics within the corpus by their expected prevalence, estimating which topics are most likely discussed by any given document in the corpus. This gives the analyst a sense of how salient some topics are versus others at the corpus level.[29]

STM differs from many other mixed membership models by specifically estimating the correlation between topics.[30] 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 with a document together (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 to identify larger themes that transcend topics. For example, painting and music may be two different topics, but they may be more likely to appear in the same "art" blogs than gun restrictions and abortion, which are more likely to appear together in "political" blogs. Every topic has some level of correlation with every other topic, but intuitively

[28]Specifically we use the semantic coherence metric developed by (Mimno et al., 2011). Our exclusivity metric is based on the work in Bischof and Airoldi (2012) and informed by Zou and Adams (2012).

[29]This quantity of interest is not unique to the STM. LDA and related models, which the STM builds off of, enables the computation of this quantity. And with ReadMe this is the only quantity of interest that can be calculated.

[30]This follows work by Blei and Lafferty (2007) on the Correlated Topic Model.

the analyst may want a test of whether the correlation is sufficiently important to merit attention. Drawing on recent literature in undirected graphical model estimation, we provide automated methods to test whether two topics are sufficiently correlated to add an edge between the two nodes. In the appendix, we describe the two graph estimation procedures we provide along with parameters set by the user.

Implementation All the features described above (estimation, uncertainty calculation, model selection) are built in to the R package `stm` and detailed in additional work [Blinded for review]. Included in the appendix of that work we also provide extensive monte-carlo simulations and data-driven permutation tests demonstrating the ability of the model to correctly recover effects without inducing false positives. That paper also focuses on the use of STM for analyzing open-ended responses in survey data and experiments, which will be of interest to comparativists.

4.1 Applications

4.1.1 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 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 interactwith 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 (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 while the rest are not. These two sources largely overlap; together, they provide expert assessments 33 of the clerics (20 Jihadists and 13 non-Jihadists) for whom Nielsen collects 11,045 texts.

We then estimate a Structural Topic Model with the binary indicator for Jihadi status as a predictor. The results are shown in Figure 4, 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.[31]

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 frequent 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 Figure 4 indicate that topics 2 (Excommunication) and 15 (Fighting) 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 ("The ruling on excommunication" by Abu Baṣīr al-Ṭarṭusī) and other rulings that make heavy use of the concept of excommunication. In contrast, topic 15 appears to be a broader Jihadist topic focused primarily on fighting the West — the ex-

[31]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.

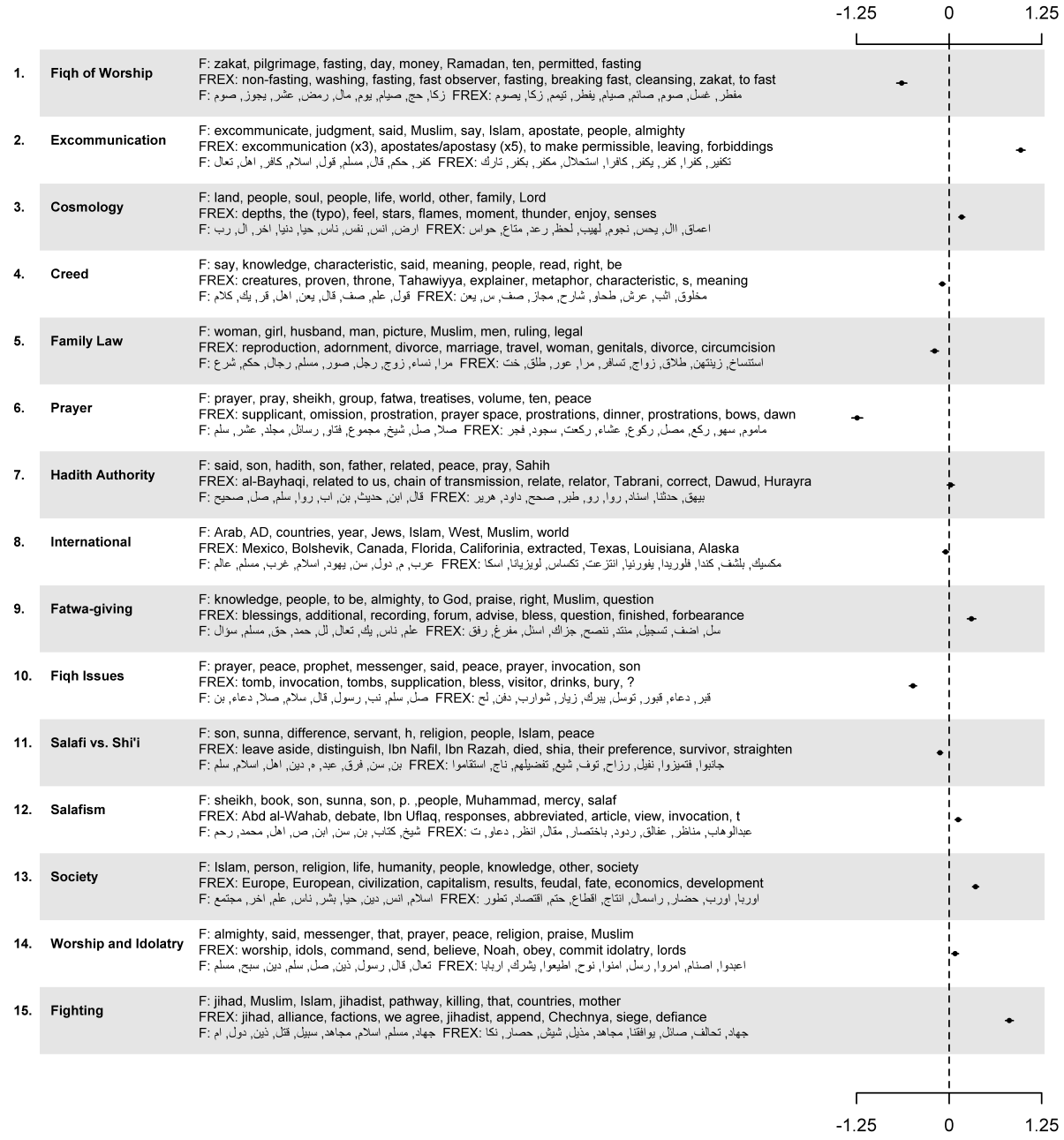

Figure 4: Coefficients and standard errors for a 15-topic Structural Topic Model with Jihadi/not-Jihadi as the predictor of topics in Arab Muslim cleric writings. The words used to label each topic are shown on the left. "F:" indicates words that are most frequent in each topic. "FREX:" indicates words that are frequent *and* exclusive to each topic. The Arabic words are in their stemmed form.

emplar documents are fatwas about the Saudi role in the Afghan Jihad, seeking military training for Jihad, and Usama bin Laden's "Letter to the fighters in Iraq and Somalia."

A number of other topics are also clearly identifiable, including topic 5 on Family Law, topic 6 on prayer, and topic 10 on assorted Fiqh Issues, including burial and invocation. 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. In Figure 5, we plot the network of topics such that topics that are correlated are linked. This shows us that there is a cluster of topics that are correlated; most of these are topics that are more likely to be used by Jihadists. In contrast, there are several topics — mostly relating to legal rulings on non-Jihadi issues — that are uncorrelated with any other topic. This tells us that among these clerics, Jihadists tend to write works that cover excommunication, fighting, salafism, international factors, and society in the same text. Similarly, clerics writing about creed are also likely to write about worship and idolatry, salafism, and cosmology. In contrast, texts about non-Jihadi legal issues — prayer, fasting, burial, family relations, and so forth — 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 while the modal non-Jihadist fatwa is paragraph-length and gives a precise ruling on only one topic.

The presence of at least two clearly Jihadist topics invites further inquiry. Figure 5 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 Figure 6. 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*.[32] Usama bin Laden has the highest proportion of words devoted to *Fighting the West* — about 65 percent — but he spends only four percent of his words

[32]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.

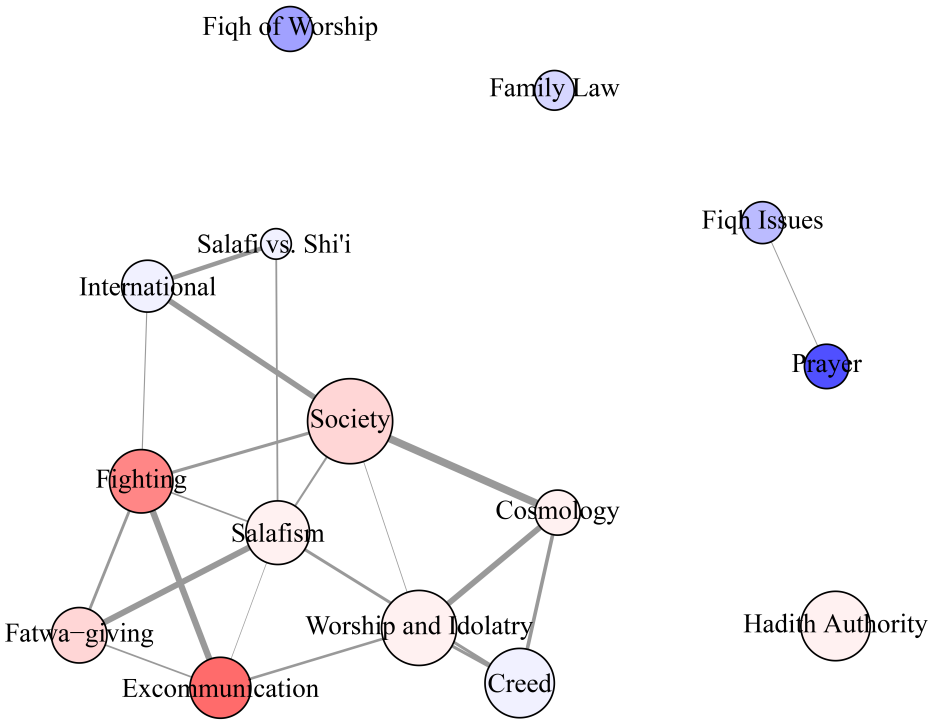

Figure 5: The network of correlated topics for a 15-topic Structural Topic Model with Jihadi/not-Jihadi as the predictor of topics in Arab Muslim cleric writings. Node size is proportional to the number of words in the corpus devoted to each topic. Node color indicates the magnitide of the coefficient, with redder nodes having more positive coefficients for the Jihadi indicator and bluer nodes having more negative coefficients. Edge width is proportional to the strength of the correlation between topics.

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 27 percent of their writing to excommunication and almost none to fighting. Abu Basir al-Tartusi tends to mention fighting more (12 percent of his words) but is similarly focused on excommunication. This is not surprising when we consider the life-trajectories of these clerics. All three have issued fatwas excommunicating prominent Muslims for alleged heresies, and Khudayr and al-Khalidi have both 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

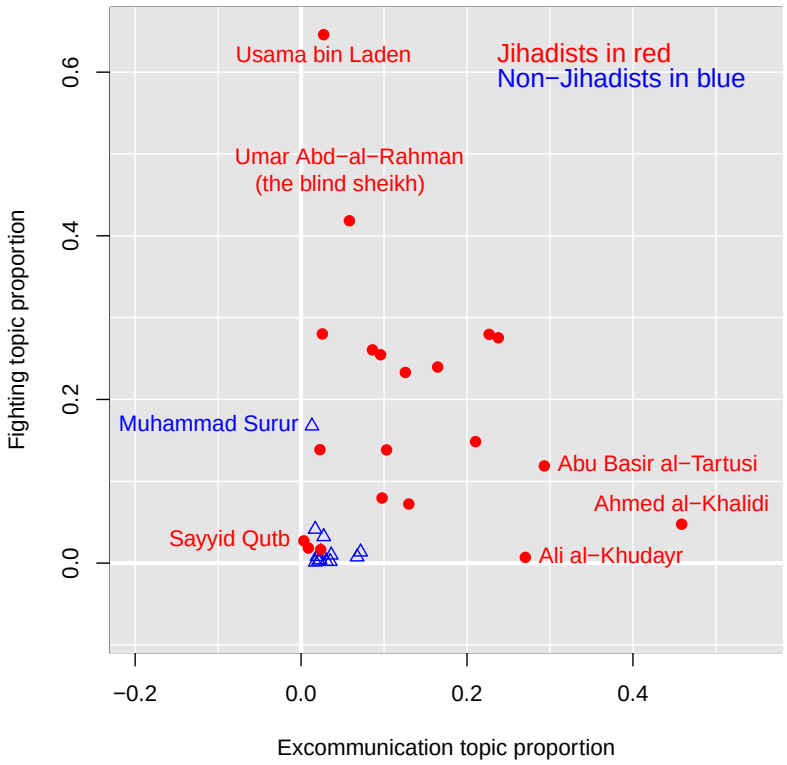

Figure 6: Estimated topic proportions by fighting the west and excommunication topics, separated out by jihadist versus jihadist coding.

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 60 percent of a document can be allocated across these Jihadi topics.

Three Jihadist authors have low enough proportions of both Jihadist topics that they could be mistaken for non-Jihadist clerics. These are not common Jihadists: Sayyid Qutb is often considered one of the founders of the Modern worldwide Jihadist movement, Muhammad Qutb is his brother, and Abu l-Ala Mawdudi is widely read and cited by modern Jihadists. 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 society, with secondary emphasis on cosmology, worship, and idolatry. More generally, we find that the current Jihadist focus on fighting the West and excommunication is relatively new. We show this

in Figure 7 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.

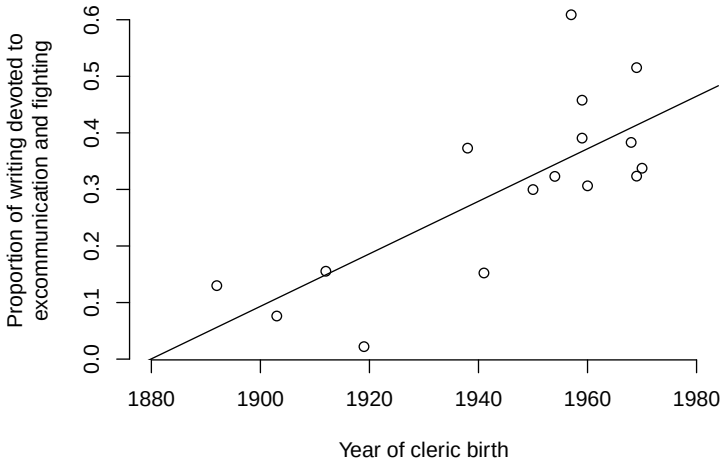

Figure 7: The proportion of words by each Jihadi author devoted to excommunication or fighting, plotted against the year of their birth with a best-fit line.

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 excommunicating fellow Muslims who are inadequately supporting the Jihadist cause and focusing on fighting the West. We also find that older writers who are considered foundational thinkers by Jihadists do not write about either of these topics, suggesting that Jihadist writing was more eclectic in the past but has become homogenized over time.

4.1.2 Newspapers and China

Our next example parallels an effort to understand how Chinese news sources cover different topics, and discuss the same topics differently [Blinded for review]. Ranging from

1997 to 2007, [Blinded for review]collected every newswire report that had the word China from five news sources (Agence France Presse, the Associated Press, British Broadcasting Corporation, Japan Economic Newswire, and Xinhua). They also collected when the newswire was released, which was coarsened to the monthly level. Hence the metadata information used in this example is time (month) and the newswire service. They then estimated a 80-topic model using the STM. The goal of the model is to uncover temporal trends in each estimated topic, as well as the influence of newswire source on how each document is discussed. The theoretical motivation is to understand how different news sources described the "rise" of China, and uncover the events that the Chinese news leaves out all together. Here we discuss several new topical time trends and differences in news sources that the model was able to discover.

First we consider change over time in the topic proportions. The STM allows for smoothing so researchers do not need to assume arbitrary functional forms linking time to topical content. Fortunately, this smoothing can detect both short and long term events. Figure 8 shows how the amount of discussion of a topic discovered by the model related to Severe Acute Respiratory Syndrome and Avian flu (H5N1) changes over time. Very little of this discussion occurs before 2003, the time of the SARS outbreak, when a huge spike occurs in discussion of this topic. A second spike occurs at the beginning of 2004 during the Avian flu outbreak. The flexibility of the time covariate allows the model to pick up short, drastic changes in discussion that are the result of international events.

The STM also allows us to measure how different news sources discuss the same topic, by allowing small changes in the distribution of words within the topic for different sources. For example, in this dataset the model discovers a topic that is related to Chinese policy toward Tibet. As displayed in Figure 9, Xinhua talks about Tibet in terms of development and natural resources, with terms like "oil", "gas", "energi", and "resourc". The Agence France Presse, on the other hand, describes Tibet with words that reflect the West's view of Chinese interference in Tibetan culture, with words such as "exil", "culture", "religion", "independence", and "buddhist". The differences in the *way* that news sources discuss a similar topic may be of direct interest to the researcher, and in addition allow the model

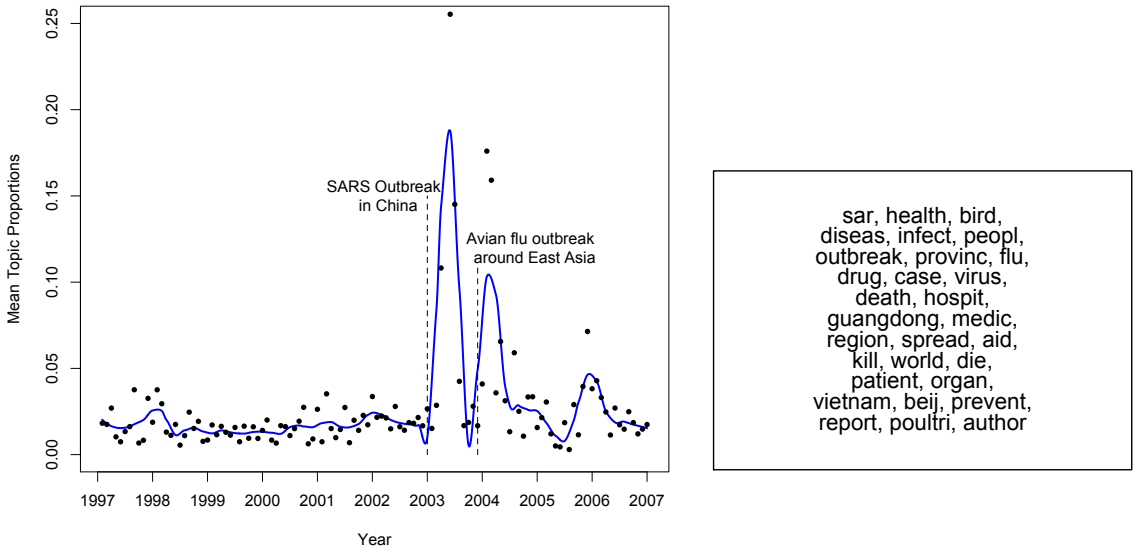

Figure 8: SARS and Avian Flu. Each dot represents the average topic proportion in a document in that month and the line is a smoothed average across time. The right hand side gives the words associated with the topic.

to account for the common feature of most corpuses: that authors have vastly different approaches to writing about the same events.

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 this data is ultimately of interest to comparativists, they will need to understand a range of issues actively being studied by scholars across multiple disciplines. And of course, there are huge volumes of text that humanity has already created, but has yet to be analyzed with some of these new technologies.

This paper 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, introduced the basics of textual pro-

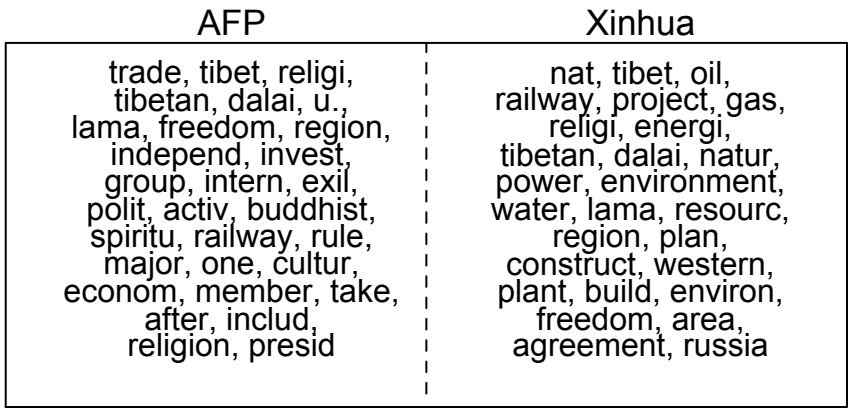

Figure 9: Discussion of Tibet, comparison of Xinhua and the Agence France Presse.

cessing with a focus on non-English texts, and then discussed techniques for text analysis, emphasizing the distinction and commonalities between supervised and unsupervised techniques. Next we introduced a new text organization framework, `txtorg`, based on the powerful Lucene engine, that not only supports many languages, but also is designed to handle large volumes of text. Finally we highlighted the Structural Topic Model and demonstrated through two examples how comparativists can use metadata to incorporate their knowledge of corpus structure into unsupervised learning. Both `txtorg` and STM are being made available as free, open-source software.

The particular structure of research problems in comparative politics (and political science more broadly) have driven the innovations we describe here. 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. Future work might also attempt to address limitations we discussed such as not taking into account word order, choosing the correct number of topics, and effective visualization of correlation structures amongst

latent topics. Finally, we emphasized that these tools do not substitute for substantive knowledge and human input, however they can provide a powerful way to structure human effort to handle the exciting and enormous volumes of political texts.

References

Alfonseca, Enrique, Slaven Bilac and Stefan 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*. Association for Computational Linguistics pp. 253–256.

Barberá, Pablo. 2012. Birds of the Same Feather Tweet Together: Bayesian Ideal Point Estimation Using Twitter Data. In *APSA 2012 Annual Meeting Paper*.

Baturo, Alexander and Slava Mikhaylov. 2013. "Life of Brian Revisited: Assessing Informational and Non-Informational Leadership Tools." *Political Science Research and Methods* 1(01):139–157.

Beauchamp, Nick. N.d. "Text-Based Scaling of Legislatures: A Comparison of Methods with Applications to the US Senate and UK House of Commons." . Forthcoming.

Bischof, Jonathan and Edoardo Airoldi. 2012. "Summarizing topical content with word frequency and exclusivity." *arXiv preprint arXiv:1206.4631* .

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

Blei, David M, Andrew Y Ng and Michael I Jordan. 2003. "Latent dirichlet allocation." *the Journal of machine Learning research* 3:993–1022.

Blei, David M and John D Lafferty. 2007. "A correlated topic model of science." *The Annals of Applied Statistics* pp. 17–35.

Blei, David M and Jon D McAuliffe. 2010. "Supervised topic models." *arXiv preprint arXiv:1003.0783* .

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

Brady, Henry E and David Collier. 2010. *Rethinking social inquiry: Diverse tools, shared standards*. Rowman & Littlefield.

Budge, Ian. 2001. *Mapping policy preferences: estimates for parties, electors, and governments, 1945-1998*. Vol. 1 Oxford University Press.

Catalinac, Amy. 2013. "Pork to Policy: The Rise of National Security in Elections in Japan.".

Chang, Jonathan. 2012. *lda: Collapsed Gibbs sampling methods for topic models.* R package version 1.3.2.
URL: *http://CRAN.R-project.org/package=lda*

Chang, Jonathan, Sean Gerrish, Chong Wang, Jordan L Boyd-graber and David M Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*. pp. 288–296.

Cheng, Kwok-Shing, Gilbert H Young and Kam-Fai Wong. 1999. "A study on word-based and integral-bit Chinese text compression algorithms." *Journal of the American Society for Information Science* 50(3):218–228.

Converse, Philip E, Warren E Miller, Jerrold G Rusk and Arthur 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.

Coscia, Michele and Viridiana 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*. ACM pp. 1412–1421.

Cutting, D, M Busch, D Cohen, O Gospodnetic, E Hatcher, C Hostetter, G Ingersoll, M McCandless, B Messer, D Naber et al. 2013. "Apache Lucene.".

Eggers, Andrew and Arthur Spirling. 2011. "Partisan Convergence in Executive-Legislative Interactions Modeling Debates in the House of Commons, 1832–1915.".

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

Feinerer, Ingo, Kurt Hornik and David Meyer. 2008. "Text Mining Infrastructure in R." *Journal of Statistical Software* 25(5):1–54.
URL: *http://www.jstatsoft.org/v25/i05/*

Gentzkow, Matthew and Jesse M Shapiro. 2010. "What drives media slant? Evidence from US daily newspapers." *Econometrica* 78(1):35–71.

George, Alexander and Andrew Bennett. 2005. *Case studies and theory development in the social sciences*. Mit Press.

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

Grimmer, Justin and Brandon M. Stewart. 2013. "Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts." *Political Analysis* 21(3):267–297.

Grimmer, Justin and Gary King. 2011. "General purpose computer-assisted clustering and conceptualization." *Proceedings of the National Academy of Sciences* 108(7):2643–2650.

Grün, Bettina and Kurt Hornik. 2011. "topicmodels: An R Package for Fitting Topic Models." *Journal of Statistical Software* 40(13):1–30.
URL: *http://www.jstatsoft.org/v40/i13/*

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

Hillard, Dustin, Stephen Purpura and John Wilkerson. 2008. "Computer-assisted topic classification for mixed-methods social science research." *Journal of Information Technology & Politics* 4(4):31–46.

Hollink, Vera, Jaap Kamps, Christof Monz and Maarten De Rijke. 2004. "Monolingual document retrieval for European languages." *Information retrieval* 7(1-2):33–52.

Hopkins, Daniel, Gary King, Matthew Knowles and Steven Melendez. 2010. "Readme: Software for automated content analysis." *Institute for Quantitative Social Science* .

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

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

Jamal, Amaney, Robert Keohane, David Romney and Dustin Tingley. 2013. American in the eyes of Arabic twitter users. In *working paper*.

Jensen, Jacob, Ethan Kaplan, Suresh Naidu and Laurence 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, Alastair Ian and Daniela Stockmann. 2007. "Chinese attitudes toward the United States and Americans." *Anti-Americanisms in world politics* pp. 157–95.

Jurka, Timothy P., Loren Collingwood, Amber E. Boydstun, Emiliano Grossman and Wouter van Atteveldt. 2013. *RTextTools: Automatic Text Classification via Supervised Learning*. R package version 1.4.1.
URL: *http://CRAN.R-project.org/package=RTextTools*

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

King, Gary, Jennifer Pan and Margaret E. Roberts. 2013. "How Censorship in China Allows Government Criticism but Silences Collective Expression." *American Political Science Review* 107:1–18.

Krippendorff, Klaus. 2012. *Content analysis: An introduction to its methodology*. Sage.

Krovetz, Robert Jeffrey. 1995. "Word-sense disambiguation for large text databases.".

Laver, Michael, Kenneth Benoit and John Garry. 2003. "Extracting policy positions from political texts using words as data." *American Political Science Review* 97(02):311–331.

Liu, Han, Fang Han, Ming Yuan, John Lafferty and Larry Wasserman. 2012. "High-dimensional semiparametric Gaussian copula graphical models." *The Annals of Statistics* 40(4):2293–2326.

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

Lowe, Will. 2013. *Austin: Do things with words.*
URL: *http://www.williamlowe.net/austin*

Lowe, Will and Kenneth Benoit. 2013. "Validating Estimates of Latent Traits from Textual Data Using Human Judgment as a Benchmark." *Political Analysis* 21(3):298–313.

Lunde, Ken. 2009. *CJKV information processing.* O'Reilly Media, Inc.

Manning, Christopher D, Prabhakar Raghavan and Hinrich Schütze. 2008. *Introduction to information retrieval.* Vol. 1 Cambridge University Press Cambridge.

McCants, Will. 2006. Militant Ideology Atlas. Technical report Combating Terrorism Center, U.S. Military Academy.

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

Mimno, David, Hanna M Wallach, Edmund Talley, Miriam Leenders and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics pp. 262–272.

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

Nielsen, Richard. 2012. "Jihadi Radicalization of Muslim Clerics.".

Nielsen, Richard. 2013. The Lonely Jihadist: Weak Networks and the Radicalization of Muslim Clerics. PhD thesis Harvard University. Ann Arbor: ProQuest/UMI. (Publication No. 3567018).

Pennebaker, James, Martha Francis and Roger Booth. 2001. *Linguistic Inquiry and Word Count: LIWC 2001.* Mahway, NJ: Erlbaum Publishers.

Porter, Martin F. 1980. "An algorithm for suffix stripping." *Program: electronic library and information systems* 14(3):130–137.

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

Ramage, Daniel, Christopher D Manning and Susan Dumais. 2011. Partially labeled topic models for interpretable text mining. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM pp. 457–465.

Ramage, Daniel, David Hall, Ramesh Nallapati and Christopher D Manning. 2009. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics pp. 248–256.

Salton, Gerard. 1989. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of*. Addison-Wesley.

Schonhardt-Bailey, Cheryl. 2006. *From the corn laws to free trade [electronic resource]: interests, ideas, and institutions in historical perspective*. The MIT Press.

Schrodt, Philip A and Deborah J Gerner. 1994. "Validity assessment of a machine-coded event data set for the Middle East, 1982-92." *American Journal of Political Science* pp. 825–854.

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

Spirling, Arthur. 2011. "US Treaty Making with American Indians: Institutional Change and Relative Power, 1784–1911." *American Journal of Political Science* .

Spirling, Arthur. 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.

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

Stockmann, Daniela. 2011. "Race to the bottom: media marketization and increasing negativity toward the United States in China." *Political Communication* 28(3):268–290.

Stockmann, Daniela. 2012. *Media commercialization and authoritarian rule in China*. Cambridge University Press.

Stone, Phillip, Dexter Dunphy, Marshall Smith and Daniel Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. The MIT Press.

Taddy, Matt. 2013. "Multinomial Inverse Regression for Text Analysis." *Journal of the American Statistical Association* 108(503):755–770.

Treier, Shawn and Simon Jackman. 2008. "Democracy as a latent variable." *American Journal of Political Science* 52(1):201–217.

Tseng, Huihsin, Pichuan Chang, Galen Andrew, Daniel Jurafsky and Christopher Manning. 2005. "A Conditional Random Field Word Segmenter." *Fourth SIGHAN Workshop on Chinese Language Processing* .

Van Atteveldt, Wouter, Jan Kleinnijenhuis and Nel Ruigrok. 2008. "Parsing, semantic networks, and political authority using syntactic analysis to extract semantic relations from dutch newspaper articles." *Political Analysis* 16(4):428–446.

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

Zhao, Tuo, Han Liu, Kathryn Roeder, John Lafferty and Larry Wasserman. 2012. "The huge Package for High-dimensional Undirected Graph Estimation in R." *The Journal of Machine Learning Research* 98888:1059–1062.

Zou, James and Ryan Adams. 2012. Priors for Diversity in Generative Latent Variable Models. In *Advances in Neural Information Processing Systems 25*. pp. 3005–3013.

A Graph Estimation

Here we describe the two estimation procedures we provide for producing correlation plots. 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.

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.