**Online Appendix**

 This appendix describes the process and various decisions that were made to derive the topics from the content of bills. For a good overview of many of these topics, see Denny and Spirling (2018).

*Modeling Issues*

We want to be confident that we have a reasonable representation of the data generating process (DGP) and that our observations are independent and identically distributed (IID). In the case of modeling the Congress, there are several issues to consider.

First, one must consider what the appropriate sample is. The dependent variable is roll call votes in the House – which are either *Yea* or *Nay*[[1]](#endnote-1) – but not all votes are created equal. In fact, many votes are procedural or amendments rather than votes that determine the passage of a piece of legislation. It is difficult to argue that these types of votes spring from the same DGP; worse, including procedural votes swamps the more salient votes that determine passage. While there are examples of salient procedural and amendment votes, there is no reason to lump different types of votes together. Accordingly, our sample consists only of final passage votes. In the 111th House there are 297 final passage votes, in the 112th there are 274, and for the 113th there are 293; together, this produces over 300,000 roll call votes.

Second, there is the issue of whether one should combine votes by members of the different political parties into one sample (controlling for party) or to treat them as two different samples. Given our restriction to final passage votes, there is an obvious temptation to increase the sample size by including both parties together. But, this flies in the face of what we know about politics. Parties in power have different considerations than opposition parties and centrist parties tend to have more complex ideologies than extreme parties (Tetlock 1984). Further, different topics may hold very different meanings depending on the party -- accordingly, the models here are disaggregate by party.

Third, it is not obvious that one should include different terms of Congress in the same sample if one wants minimally IID observations. Agendas exist and they vary over time (and not in some nicely continuous fashion). Votes are always about something and there is no reason to believe that these votes are drawn from the same DGP in different terms, especially if there is a change in party control of the agenda.[[2]](#endnote-2) Indeed, there is every reason to think the agenda matters:

It is something of an open question whether any session of Congress would prove to be typical. All would have entirely usual aspects and some very unusual ones. As one congressman remarked, ‘There hasn’t been a year since I’ve been here that has been a usual year.’ (Kingdon, p. 15).

For the terms considered here, there is also abundant evidence that strategies and the substance of legislation changed markedly from term to term. Republican Speaker John Boehner’s attempts to enforce the Hastert rule was in contrast to the strategy adopted by his Democratic predecessor, Nancy Pelosi. The solution to this problem is to simply fit each term of Congress separately. This is the also the approach taken by Gerrish and Blei. Treating terms independently substantially improves results.

Last, we could generate topics using STM for each individual Congress separately or by aggregating our data into one corpus. We have done both, but given the demands (and number of parameters) of topic models, we achieve better results by aggregating the data. Using the 112th as an example, we lose approximately 6% accuracy for the minority party when generating separate topic models. It must also be noted that generating topics from one corpus (as is done by prior efforts in this vein) violates in part the spirit of forecasting when topic models must deal with new terms in the dictionary. We have an alternative approach that re-estimates the topic model whenever new terms arrive; results are nearly identical to those found in this paper and are available on request.

*Why use topics as features?*

Using text as data has become much more important in the social sciences.[[3]](#endnote-3) Our approach depends on extracting topics from the full text of bills before Congress. But, it is important to discuss why feature extraction of this kind is important in the context of predicting votes. After all, if one wants to intervene as little as possible between the data and the model, one could instead build a statistical model that uses word counts from each bill directly. While the total dictionary size for a modern term of Congress is nearly 16,000 unique words (thereby producing an equal number of IVs), regularization approaches like the lasso could trim this space substantially.

As noted in the main text, feature selection is a cornerstone of machine learning and serves to select signal over noise in the data (Bishop, 1995). To see why feature selection via topic models (LDA or STM) performs so well, we consider one alternative which is to instead use the raw word counts in each bill directly. Accordingly, we created a dataset based on the complete dictionary of each term of Congress, resulting in nearly 16,000 IVs (one column per token). To push coefficients which contributed little to the predictive power of our model to zero, we relied on regularized regression approaches like the lasso.[[4]](#endnote-4) The lasso is a simple amendment to a regression model that penalizes model complexity. In addition to minimizing the residual sum of squares, an extra term is added to the minimization that is the sum of the absolute value of the coefficients (Tibshirani 1996). One issue, however, immediately presents itself with using word counts directly. Regularized regression produced stable models with this dataset when we generated out-of-sample results for an entire term using k-fold cross-validation. But, for the genuinely predictive task of forecasting votes in the 113th Congress as each bill occurs, there are too few rows (a maximum of 435) available.

 For both the 111th and 112th terms, regularized regression models produced accuracies just over 0.8.[[5]](#endnote-5) This is substantially worse than regressions based on topic models and reinforces the need for pre-processing and feature extraction in problems of this kind.

 It is also worth noting that feature selection, in this area, only takes us so far. Attempts to produce a simpler representation of bills than our topic model or to map substantive topics to ideology (as measured by NOMINATE’s two dimensions) were not fruitful. As noted in the main body of the text, we attempted to find simpler representations of topics using both linear (i.e., PCA) and non-linear mappings (i.e., a variational autoencoder). Using the 113th Congress as an illustration, a PCA performed on all 50 topics in our model plus the first two dimensions of NOMINATE produced 23 latent variables with an eigenvalue of one or above; the eigenvalue of the first dimension was only 2.9 Worse, the first 23 factors explained only 64% of the variance of the data and neither NOMINATE dimension was well described by these factors. The 111th and 112th produced worse results and none of the latent variable representations of our topics performed well in forecasting models. The bottom line is that the topics are distinct both from each other and from measures of left-right ideology. And, to successfully forecast votes, one needs relatively complex topic models of bill content using a minimum of fifty dimensions.

*Parameters for Topic Models*

Building topic models from the text of bills is a complicated process given the diversity and frequency of idiosyncratic and technical terms in legislation. Following Gerrish and Blei (2011), we used TreeTagger (Schmid 1994) to stem and lemmatize bills, called from the koRpus package in R.[[6]](#endnote-6) Unlike Gerrish and Blei, we did, however, improve the results by hand-coding unknown tokens that represent specialist vocabulary commonly found in bills. For example, the word “Medicaid” occurred 3077 times in the 111th House alone; “cybersecurity” appeared 556 times. The resultant topics were substantially improved by including these terms.[[7]](#endnote-7)

Our corpus is the set of all bills before each term of Congress. Bills consist of a vector of words **w** of length N, where each word is the result of pre-processing. It is worth noting that LDA and STM employ a bag of words (exchangeability) assumption; the order of the words in **w** is unimportant in the statistical model.

Given this, LDA estimates a number of topics based on the corpus; this number is a parameter chosen by the modeler and here is set at 100. Each bill d is expressed as a vector **θ**d consisting of a proportion for each of the one hundred topics that could make up a bill.

Following Blei, et al. (2003), first choose a number of words N in a bill:

N ~ *Poisson*().

Second, the proportions of topics are drawn from a common prior with shape parameters α:

**θ** ~ *Dirichlet*(αθ).

Last, for each of N words in the dictionary built from the bills in a term, we iterate the process of choosing a topic and then within the topic the word that is represented. For each word wn from a bill d, a topic z is drawn as z *~ Multinomial*(**θ**d); words are drawn from p(wn|z,β), a multinomial distribution of word probabilities for that topic**.** Topics are nearly independent (since they sum to unity) in LDA and the number of topics is fixed prior to estimating the model.

The parameter space implied by this model is huge – each of the N topics used is a vector of weights over the entire dictionary of words, which even after aggressive pre-processing has between 5 thousand and 10 thousand entries for any given term of Congress. Unsurprisingly, multiple local optima exist for the likelihood function implied by this model. Thus, to the extent that researchers are presenting word clouds or lists of the most salient words by topic, often these results are non-unique and as such should be taken with a grain of salt.

The solution we employ, outlined by Roberts, Stewart, Tingley, and Airoldi (2014) extends topic models in two important ways. First, the STM model allows for spectral learning, which provides outcomes consistent with the global optimum. This requires an additional assumption of separability (identified by Arora, et al. 2013), where a unique anchor word is identified for each topic in **θ**.

Second, the STM model allows for covariates at two levels: topic prevalence and the content of topics. If, for example, multiple terms of Congress are aggregated into one dataset and party control of the agenda changes, it is necessary to consider that a Republican authored bill using the word “women” might be entirely different in meaning than a Democrat authored bill using “women”. Though we do not focus on results generated by aggregating terms here, we relied on STM’s ability to use a covariate for topic prevalence – in this case, party control – to deal with this issue.[[8]](#endnote-8) For topic prevalence covariates **x**,

**θ**d ~ LogisticNormal(**Γ**’**x**’d, topics).

If no covariates are used, STM is identical to the correlated topic model (Blei and Lafferty 2007), which relaxes LDA’s assumption of independent topics but is otherwise the same. STM is a meaningful generalization of topic models that allows known attributes of each bill to influence topics. For these reasons, we use the STM package in R to estimate topics. For redundancy, we also generated results with LDA implemented by the python **gensim** library.[[9]](#endnote-9)

Another concern with topic models is how one should choose parameters, including the number of topics and the lower and upper frequency threshold for including tokens in the model. We ran simulations using two training sets, one from the 111th term and another from the 112th. We evaluated performance for the following combinations:

Number of topics ∈ {10, 25, 50, 100}

Lower threshold (count) ∈ {5, 10, 25, 50}

Upper threshold (frequency) ∈ {.2, .3, .5}

This produces a total of 96 possible topic models and we evaluated each in terms of out-of-sample accuracy. Results in both terms were nearly identical. Weakly, the higher the lower count threshold the better; the higher frequency threshold had no effect. The number of topics, though, mattered a great deal and overall, more topics produced higher accuracy (as represented in the figure below):



We settled on 100 topics, a lower count threshold of 25, and an upper frequency threshold of .5 for the models presented in the main text in the out-of-sample tests using the 111th and 112th terms. For the predictive model using the 113th term, we used 25 topics instead which was necessary for convergence.

Finally, we experimented with multiple different pre-processing schemes for the corpus. In general, the only choice that had a substantial impact on our models was replacing “unknown” tokens (e.g., we would normally exclude “Dodd-Frank” as proper names, but the usage in this case demanded inclusion in the dictionary) by hand. We also found that bigrams gave the biggest bang for the buck in terms of the tradeoff between accuracy and increasing the size of the parameter space.

*Additional measures of model fit*

We also present additional information on model fit, specifically the area under the ROC curve and graph for each of our datasets. Our focus here is on out-of-sample fit for minority party votes, given the ease with which one can predict votes for the majority.

|  |  |
| --- | --- |
| **Congress** | **AUC** |
| 111th | 0.927 |
| 112th | 0.980 |
| 113th | 0.988 |



111th Republicans AUC, 112th Democrats AUC, and 113th Democrats AUC

1. There can also be a value of “not voting.” One thus has the two main options of either treating these observations as missing or as a *Nay* vote – we settle on the latter for the work presented here but results are robust to coding the values as missing instead. [↑](#endnote-ref-1)
2. There are other ways to deal with time – see Wang, et al. (2013) for an alternative approach. Note also that there is a distinction between the regression stage in our modeling process (which should be disaggregated by term) and generating the topic model (which should not be disaggregated given the data requirements of LDA / STM). [↑](#endnote-ref-2)
3. A partial list of research using text as data is Laver, et al. (2003), Lauderdale and Clark (2012); Grimmer and Stewart (2013), Roberts, et al. (2014), and Kim, et al. (2014). [↑](#endnote-ref-3)
4. We also used the ridge penalty and an elastic-net penalty of 0.5. None of these models performed as well as topics-based regression models. [↑](#endnote-ref-4)
5. Random forest models did better but did not exceed 0.88. [↑](#endnote-ref-5)
6. http://r.reaktanz.de/pckg/koRpus/koRpus.pdf [↑](#endnote-ref-6)
7. TreeTagger provided us with lemmas which we used to build the dictionary for bills before Congress – this helped reduce the size of the dictionary. It also provided parts of speech, which we did not use in our model. TreeTagger’s default dictionary lacked technical terms, salient proper names, and modern words though which were commonly found in Congressional bills. Additional examples include “Cherokee”, “subprime”, “Dodd-Frank” (after forming bigrams), and “permittee”. In these cases, TreeTagger marked the word as unknown and we manually added in the proper lemma afterwards. There is a large literature on improving lemmatizing for technical contexts, but the bottom line is that one should probably not use “off-the-shelf” tools without modification (Liu, et al. 2012). [↑](#endnote-ref-7)
8. Aggregating terms produces worse results during periods when party control changes. For a simple comparison with the results here, our model that aggregated the 111th and 112th terms of the House had an out-of-sample accuracy rate of approximately 90% using the STM model. Using the simpler LDA model did less well – approximately 88%. [↑](#endnote-ref-8)
9. In the case of LDA, we found that multiple models of single terms provided the same fit to the data albeit with very different topic – word mappings. The overall fit of the regressions predicting roll call votes were the same with LDA when we disaggregated by Congress. STM does, however, have a significant advantage when it comes to replicability. [↑](#endnote-ref-9)