**Structural Topic Models for Open Ended Survey Responses**

*American Journal of Political Science*

# 1 Online Appendix

This appendix outlines a number of important details about the STM. We have organized it into three sections (1) Model Estimation, (2) Model Validation, and (3) Getting Started. The Model Estimation section (1.1) presents the details of model estimation, which describes an Expectation-Maximization based approach to optimization of the model parameters.

With this formal statement in mind, the Model Validation section overviews theoretical and simulation evidence for the properties of the STM. Section 1.2.1 discusses the STM compared to alternative tools for text analysis, highlighting both similarities as well as important points of departure. Next in Section 1.2.5 we turn to a set of Monte Carlo based evaluations of the STM. We also include a discussion of inference in one of the paper's empirical examples using a two-step LDA based analysis that does not formally incorporate covariates in estimation.

The final section includes information directed at the applied user. Section 1.3.2 discusses the multiple testing problem and the use of the STM with pre-analysis plans. Section 1.3.4 gives an example of using the STM alongside mediation analysis. Section 1.3.1 demonstrates two-software tools we have developed to enable efficient work-flow for researchers using our methods.

## 1.1 Model Estimation

Before providing an overview of estimation we give the data generating process:

1. Draw $\vec{\eta}_d | X_d\gamma, \Sigma \sim \mathcal{N}(\mu = X_d\gamma, \Sigma)$

2. For $n \in 1, \ldots, N_d$:

   - Draw topic assignment $Z_{d,n}|\vec{\eta}_d$ from Multi($\theta = \frac{\eta_d}{\sum_\eta}$))
   - Draw word $W_{d,n}|z_{d,n}, \vec{\kappa}$ from Mult($\beta \propto \exp(m + \kappa_k z_{d,n} + \kappa_c U_d + \kappa_I U_d z_{d.n})$

There are two substantive differences to the notation from the main paper. We leave the topic proportions in their unnormalized form and we expand out the notation for prior distribution over words. Note also that $\gamma$ and $\kappa$ are maximized with a user-specified regularizing prior. The defaults in the forthcoming `R` package are a Normal-Gamma prior for $\gamma$ and a sparsity promoting Gamma-Lasso (Taddy, 2013).

Model estimation proceeds via semi-collapsed mean-field variational EM. We integrate out of the token-level assignments $z$ and infer the unnormalized document proportions $\eta$ during the variational E-step. Then we solve for the token-level variational posterior $\phi$ in closed form. The global parameters are inferred in the M-step. Estimation is complicated by the non-conjugacy of the logistic normal and the multinomial. To deal with non-conjugacy, in the variational E-step we use a second-order Taylor series approximation to the optimal collapsed variational update as in Wang and Blei (2013). Note that this means inference in our model does not enjoy the theoretical guarantees of mean-field variational inference in the conjugate exponential family (see for example Grimmer (2011) for an introduction to conjugate variational approximations for political scientists). Nonetheless, the approximation described here has been shown to work well in practice in numerous applications as well as in evaluations by Wang and Blei (2013).

Full details are available in a companion technical manuscript and code will be available on Github upon publication, if not before. However, for completeness we overview the estimation below, omitting derivations.

### 1.1.1 General Inference Strategy

In mean-field variational bayes, we form a deterministic factorized approximation to the posterior which minimizes the KL-divergence between the true posterior and the approximation (Bishop, 2007; Grimmer, 2011). This turns the Bayesian inference problem into an optimization problem. Thus we avoid the complications of convergence monitoring in MCMC and also typically enjoy substantially faster run times.

In our case, we approximate the posterior distribution over the document-level latent variables $p(\eta, \vec{z})$ with the factorized posterior $q(\eta)q(\vec{z})$ which given the conditional independence assumptions of the model further factorizes as $q(\eta) \prod_n q(z_n)$. We then seek to minimize the KL-divergence to the true posterior, KL($q||p$). In conjugate exponential family cases, standard derivations (Bishop, 2007) show that the optimal updates for the variational posterior are:

$$q^*(\eta) \propto \exp(E_{q(z)}[\log p(z, \eta)]) \tag{1}$$

$$q^*(z) \propto \exp(E_{q(\eta)}[\log p(z, \eta)]) \tag{2}$$

Since the logistic-normal is not conjugate to the multinomial these distributions will not be an easily normalized form. Instead we use an approximation (described in more detail below) such that $q(\eta)$ will be multivariate normal distribution with mean $\lambda$ and variance matrix $\nu$, and $q(z)$ will be discrete with parameter $\phi$.

We speed convergence by integrating out $z$ before optimizing $q(\eta)$. This allows us to skip the expensive iterative updating process at the document level by finding the parameters $q(\eta)$ that jointly maximizes the local document posterior.

The global parameters $(\gamma, \Sigma, \kappa)$ are set to their MAP estimates during the M-step. Thus the inference algorithm is analogous to a standard EM algorithm, except that we take the expectations with respect to the variational distributions $q(\eta)$ and $q(z)$.

Thus in each stage of the EM algorithm we go through the following steps (note we have placed substantive interpretations of the parameters in parentheses):

- For each document
  - Update $q(\eta)$ (document-topic proportions) by optimizing the collapsed objective
  - Solve in closed form for $q(z)$ (assignments to topic for each word)
- Update $\gamma$ (coefficients for topic prevalence)
- Update $\Sigma$ (global covariance matrix controlling correlation between topics)
- Update each $\kappa$. (topical-content parameters describing deviations from baseline word rate)
- Repeat until convergence.

Convergence can be assessed by monitoring either an approximation to the bound on the marginal likelihood, or by change in the variational distribution $q(\eta)$. In practice we monitor convergence using proportional change in the approximate bound on the marginal likelihood at the global level as in Wang and Blei (2013).

### 1.1.2 E-Step

In our case, the necessary expectations are not tractable due to the non-conjugacy, and so we use a Laplace approximation. Following **?** we find the MAP estimate $\hat{\eta}$ and approximating the posterior with a quadratic Taylor expansion. This results in a Gaussian form for the variational posterior $q(\eta) \approx \mathcal{N}(\hat{\eta}, -\nabla^2 f(\hat{\eta})^{-1})$ where $\nabla^2 f(\hat{\eta})$ is the hessian of $f(\eta)$ evaluated at the mode.

Integrating out the token level latent variables we solve for $\hat{\eta}$ for a given document which amounts to optimizing the function,

$$f(\hat{\eta}) = -\frac{1}{2}\log|\Sigma^{-1}| - \frac{K}{2}\log 2\pi - \frac{1}{2}(\eta-\mu)^T\Sigma^{-1}(\eta-\mu) + \left(\sum_v c_v \log\sum_k \beta_{v,k}e^{\eta_k} - W\log\sum_k e^{\eta_k}\right) \tag{3}$$

where $c_v$ is the count of the $v-$th word in the vocabulary and $W$ is the total count of words in the document. This gives us our variational posterior $q(\eta) = \mathcal{N}(\lambda = \hat{\eta}, \nu = -\nabla^2 f(\hat{\eta})^{-1})$.

Once $q(\eta)$ is updated, $q(z)$ is easily updated as a function of the mean of $q(\eta)$, $\lambda$, by

$$\phi_{n,k} \propto exp(\lambda_k)\beta_{k,w_n} \tag{4}$$

### 1.1.3 M-Step

In the M-step we update the global parameters which control topic prevalence and topical content. The parameters for topical prevalence, are simply penalized maximum likelihood estimation for a penalized seemingly unrelated regression model.

$\Sigma$ is estimated as in Blei and Lafferty (2007) with the exception that each document now has its own mean via $X_d\hat{\gamma}$ and each document has a full covariance matrix $\nu_d$. This yields the update,

$$\hat{\Sigma} = \frac{1}{D}\sum_d \nu_d + (\lambda_d - X_d\hat{\gamma})(\lambda_d - X_d\hat{\gamma})^T$$

In the estimation for topical content, we base our estimation strategy off of the work of Eisenstein *et al.* (2011), again doing MAP estimation. The idea is that each topic is represented as a deviation from the baseline word frequency $m$ in log-space. Thus for topic $k$, the rate is $\exp(m + \kappa_k)$. Conditional on the token assignments $\phi$ this is a multinomial logistic regression on words with a fixed intercept.

When estimating with covariates, for each covariate $j$ we consider the word rate deviation for any document of that covariate $\kappa_j$ as well as a topic-covariate interaction $\kappa_{j,k}$. Thus the form for the word probabilities is,

$$\beta_{k,j} \propto \exp(m + \kappa_k + \kappa_j + \kappa_{j,k}) \tag{5}$$

We then solve each block of the $\kappa$'s separately with objective functions analogous to the one above.

#### 1.1.4 FREX and Semantic Coherence Scoring

In this section we provide technical details for a few of our model diagnostics. FREX words are a way of choosing representative words for analysis. It builds off the work of Bischof and Airoldi (2012) but with a substantially simpler model. Specifically FREX is the geometric average of frequency and exclusivity. We define the exclusivity term as:

$$\text{Exclusivity}(k, v) = \frac{\beta_{i=k,j=v}}{\sum_i \beta_{j=v}}$$

where $i$ indexes the topics and $j$ indexes the vocabulary words. Then our FREX score is:

$$\text{FREX}(k, v) = \left( \frac{\omega}{\text{ECDF}(\text{Exclusivity}(k, v))} + \frac{1 - \omega}{\text{ECDF}(\beta_{k,v})} \right)^{-1}$$

where ECDF is the empirical CDF function, and $\omega$ is the weight given to exclusivity, here .5 by default.

Semantic coherence comes from Mimno *et al.* (2011) and is based on co-occurrence statistics for the top $n$ words in a topic. Thus where $D()$ is a function of a word index $v_i$ which outputs the number of documents containing its arguments, we get

$$\sum_{n=2}^{N} \sum_{m=1}^{n-1} \log \left( \frac{D(v_n, v_m) + 1}{D(v_m)} \right)$$

Intuitively this is a sum over all word pairs in the top topic words, returning the log of the co-occurrence frequency divided by the baseline frequency. The one is included to avoid taking the log of zero in the event that a pair of words never co-occurs (which is possible with very short documents). This measure is closely related to pointwise mutual information.

### 1.1.5 Uncertainty

We can incorporate estimation uncertainty into subsequent analyses using the method of composition. Popularized in political science by Treier and Jackman (2008) the central idea is to integrate over our estimated posterior. We draw a sample from each document's approximate posterior as $\tilde{\eta}_d \sim \mathcal{N}(\lambda_d, \nu_d)$ and convert to the simplex $\tilde{\theta}_d = \frac{\exp(\tilde{\eta}_d)}{\sum_k \exp(\tilde{\eta}_d)}$. Then we calculate our standard regression model and simulate from the regression's posterior. We repeat this process 100 to 1000 times and then combine all the results together to form an approximate posterior which combines our estimation uncertainty. In ourR package we provide tools both for simulating from the approximate posterior over $\theta$ and calculating uncertainty under common regression models.

## 1.2 Model Validation

This section of the appendix overviews a series of validation exercises including comparisons to existing unsupervised models. Understanding that no single test can validate the model, we build confidence in our approach by marshalling evidence from tests using simulated data, permutation tests and additional tests using the real documents in our presented experiments. When using real documents, we use the immigration dataset from Gadarian and Albertson (2013) throughout for continuity.

In Section 1.2.1 we compare the STM to prominent unsupervised models within the literature, highlighting the contrasts to existing approaches. We discuss the merits of including covariates and jointly estimating their effects on the topics rather than using a two-step approach. In Section 1.2.5 we use simulations to demonstrate that the STM is able to recover parameters of interest. In Section 1.2.6 we move from simulated data to a permutation test on actual documents in which we randomly permute the treatment indicator to demonstrate that the STM does not find spurious treatment effects. In Section 1.2.7 we provide a comparison with a two-stage approach using standard LDA, highlighting the differences on the immigration data. Finally, in Section 1.2.8 we compare the STM results on the immigration data to the analysis of the same data with human coders.

### 1.2.1 Comparison to Alternative Models

Computer-assisted methods for the measurement of political quantities of text have already seen widespread use in other areas of political science (Laver *et al.*, 2003; Slapin and Proksch, 2008; Grimmer, 2010; Quinn *et al.*, 2010).[1] In this section we contrast our approach with three alternative unsupervised text analysis models in the literature, focusing on the advantages of including covariates.

### 1.2.2 LDA and Factor Analysis

The standard Latent Dirichlet Allocation model can provide rich, interpretive semantic depictions of text. However, central to LDA is the assumption of *exchangeability*, the idea that the ordering of the documents in the corpus is irrelevant. Not only do we believe that survey respondents naturally have systematic, and easily-measurable, differences (e.g. gender, income etc.) but in experimental settings the design of the study suggests non-exchangeability between treatment and a control condition. We could always simply ignore this and run the standard LDA model, but even in a best case scenario where comparable topics are estimated, the resulting estimate of the covariate effects will be inefficient. We describe the process of running LDA (without covariate information) followed by analysis of covariate relationships with the results resulting topics as a "two-step" approach.

Indeed, Hopkins (2012) provides a recent example of the best use of the two-step approach. He uses the correlated topic model, a close relative of LDA, to study open-ended survey responses from 30,000 respondents on healthcare. One of his primary objectives

[1] In restricting our focus to the political science literature we naturally omit the large literature in computer science, statistics and other social sciences. See Blei (2012) for a review of Latent Dirichlet Allocation and related models.

is to show trends over time and reaction to elite framing in press releases. Models such as the STM would allow for an analysis that directly incorporates the effects of time on topic proportions and also makes use of the rich demographic meta-data available on the individual responses.

In Simon and Xenos (2004), the authors propose the analysis of open-ended survey response using latent semantic analysis in a three stage process: data preparation, exploratory factor analyses and hypothesis testing. Specifically they advocate the use of latent semantic analysis (LSA) because it is "rigorous, systematic, and, most of all, efficient." We agree with the goals of their work. Indeed our model exists in a lineage that extends from latent semantic analysis; LDA was developed as a probabilistic formulation of latent semantic analysis which is appropriate to discrete data such as word counts, the STM is a related alternative to LDA appropriate to the inclusion of rich covariate information.[2] While LSA was amongst the best available techniques for the dimension reduction of text at the time of the article, the STM represents advances in statistical rigor and efficiency of the intervening decade. Furthermore, because The STM produces an approximate posterior distribution over topic classifications, it is more amendable to the kind of hypothesis testing advocated by Simon and Xenos (2004). By propagating our uncertainty about a document's coding, we can lower the risks of measurement error.[3]

We also note that the clear data generating process of the STM, allows us to avoid the numerous practical issues that are discussed in length by Simon and Xenos (2004) such as, weighting schemes of the word-frequency matrix and selection of rotation. The topics themselves are also more interpretable than factor loadings, specifically in the STM or LDA, the topic loading for a document is the percentage of words attributable to a latent topic; by contrast, eigenvalues and document scores have little *prima facie* meaning to

[2] Buntine and Jakulin (2005) shows that multinomial principal components analysis and LDA are closely related models. In addition to the numerous theoretical benefits, the use of the LDA framework side steps the numerous practical difficulties of the standard principal components analysis advocated by Simon and Xenos (2004). In particular the user is not faced with un-interpretable quantities such as negative factor loadings or concerns about rotations.

[3] See Grimmer (2010) for arguments on this in the context of text data, and Blackwell *et al.* (2011); Treier and Jackman (2008) on measurement error in other contexts.

the user. Thus, we argue that the STM offers all the advantages of the framework laid out by Simon and Xenos (2004) with the addition of numerous quantities of interest.

### 1.2.3 Single-Membership Models with Covariates

The literature on single-membership models in political science offers another appealing approach. Work such as the Dynamic Topic Model (Quinn *et al.*, 2010) and the Expressed Agenda Model (Grimmer, 2010), take into account systematic variation amongst documents based on *specific* characteristics such as time and authorship. These models allow the analyst to specify political knowledge about the documents at the outset. However, these existing single-membership models have focused on modelling particular types of structure. The STM generalizes these models by allowing any structure which can be captured in a general class of linear models. This allows the STM to model time, authorship, or time and authorship interactions all within the same framework. The STM also provides variational posteriors over the parameters which provide estimates of uncertainty (similar to the Expressed Agenda Model but not the Dynamic Topic Model).

These models also assume that responses contain words originating from only a single topic. This can be advantageous as it often makes the corresponding optimization problem easier, with fewer local modes. However, we argue that in most open-ended responses, survey respondents are likely to give multi-faceted answers which could touch on a variety of semantic themes. At a minimum it is desirable to relax this restrictive assumption and see if the results change substantively. Furthermore, each of these models assume that topics are talked about in the same way, whereas as the STM allows (for example) men and women to talk about the same topics differently, as we describe in our empirical examples and simulations.

Single-membership models in political science do establish and excellent tradition of validation of the topics based on external political phenomena and careful reading (Grimmer and Stewart, 2013). These validations can be performed in our setting by using model output to examine documents which contain a large percentage of words devoted to a particular topic. In the STM model, the analyst can also use the covariate relationships in both topic prevalence and topical content to verify that the results are sensible and in line with theoretical expectations. The best validations will necessarily be theory-driven and application specific, but we direct interested readers to Krippendorff (2004) for some excellent guidelines and Quinn *et al.* (2010) for some excellent applied examples.

Thus, previous work highlights a useful insight; including information about the structure of documents can improve the learning of topics. These changes can be motivated in the language of computer science via the no-free lunch theorem (Wolpert and Macready, 1997), or statistically via literature on partial pooling (Gelman and Hill, 2007). The source of the improvement is the identification of units which are "similar" and allowing the model to borrow strength between those units, without making the stronger exchangeability assumptions.

We can summarize the relative merits of the STM framework with Table 1 inspired by the table of the common assumptions and relative costs of text categorization in Quinn *et al.* (2010).

| | Hand Coding | Factor Analysis | LDA | Single Member | STM |
|---|---|---|---|---|---|
| Categories Known | Yes | No | No | No | No |
| Mixed-Membership | No | Yes | Yes | No | Yes |
| Covariates on Prevalence | No | No | No | Limited | Yes |
| Covariates on Content | No | No | No | No | Yes |
| Interpretable QOIs | Yes | No | Yes | Yes | Yes |
| Uncertainty Estimates | No | Limited | Yes | Yes | Yes |

Table 1: Relative Merits of STM and competing approaches for analysis of open-ended survey response.

### 1.2.4 Estimation vs. Two-Stage Process

We can contrast the STM's joint estimation of covariates and topics with a two-stage process in which some method (factor analysis, LDA, human classification) is first run and then the estimated topic proportions are entered into a regression of interest. The intuitive appeal of a two-stage estimation with LDA is the ease of use and the separation of hypothesis testing from measurement. Indeed the analyst might reasonably be concerned that joint estimation of the topics and covariates will induce spurious correlations between the topics and the variables of interest (although in the next section we demonstrate using

Monte Carlo simulations that this is not the case).

The hidden cost of two-stage estimation is the inability to adequately account for our uncertainty. Specifically, we are unable to propagate the uncertainty in our classifications to our regressions with covariates, since the naive approach will condition on the observed values of the topics developed in the first stage. Furthermore, the model implicitly assumes that, in expectation, the topics are discussed in the same way and with the same prevalence across all respondents. In Section 1.2.7 we explore this approach and highlight problems with it.

The concern about spurious correlations is reasonable, and any time we use unsupervised methods for measurement we must be careful to validate our findings. Spurious correlations can arise even in a two-stage process if, for example, the analyst iterates betweens specifications and observing covariate relationships. In the STM model, we address these concerns with regularizing priors which draw the influence of covariates to zero unless the data strongly suggests otherwise. In practice, this means that the analyst can specify the degree to which she wants the covariates to enter into the model. Standard diagnostics such as posterior predictive checks or cross-validation can also be used. Posterior predictive checks in particular have been shown to be able to diagnose violations of assumptions in LDA, providing one method of checking whether additional structure needs to be incorporated in the model (Mimno and Blei, 2011).

### 1.2.5 Monte Carlo Analysis

In this section we use simulated data to test the model. Unfortunately the process of writing natural language text bears little resemblance to the bag-of-words data generating process assumed by almost all topic models. In some cases this is to the model's disadvantage, because the interdependencies amongst words are more deeply complex than the model's assumptions. Yet, in many cases it is an advantage; there are features to natural language text, that make it amenable to analysis of this sort: people write with finite vocabularies, use words consistently and tend to write about relatively focused topics. Simulation parameters must be carefully chosen to reflect patterns we can reasonably expect to find in text. Accordingly we provide a series of tests in this and the following sections which move from purely simulated documents to full analyses on real texts.

The advantage of purely simulated data is that we can compare recovery to the known truth. The evaluation of parameter recovery is complicated by issues of identification in the model. At the most basic level topics are invariant to label switching (Topic 1 and Topic 2 could be switched since the number is irrelevant), and there can be myriad causes of local modes in the posterior. Indeed under some very general conditions the LDA inference problem is probably NP-hard (Sontag and Roy, 2009). In order to compare comparable units between model fits, we use the Hungarian algorithm to approximate the best possible match (Papadimitriou and Steiglitz, 1998; Hornik, 2005). The Hungarian algorithm solves the optimal assignment problem (which is typically NP-hard) in polynomial time, a speed advantage which becomes important when the number of topics is above 4.

**Basic Simulations** For our most basic simulations, documents are created according to the generative model for LDA with different means for each topic depending on the treatment assignment. Thus the generative process for the corpus parameters is:

$$\begin{aligned} \beta_k &\sim \text{Dirichlet}(.05) \\ \alpha_{t=0} &= (0.3, 0.4, 0.3) \\ \alpha_{t=1} &= (0.3 - ATE, 0.4, 0.3 + ATE) \end{aligned}$$

where each of the $k$ topics for $k \in 1, 2, 3,$ is drawn from a symmetric 500-dimensional Dirichlet with concentration parameter of .05. The mean of the document topic proportions is based upon the treatment assignment, where the first half of the documents are assigned control, and the second half are assigned treatment. Then for each document

$$\begin{aligned} N_d &\sim \text{Poisson}(\zeta) \\ \theta_d &\sim \text{Dirichlet}(\alpha_d * G_0) \\ \vec{w}_d &\sim \text{Multinomial}(N_d, \theta_d \beta) \end{aligned}$$

We consider a host of different parameters but for space present here only the results across the following dimensions: Number of Documents (100 to 1000 by 100 increments),

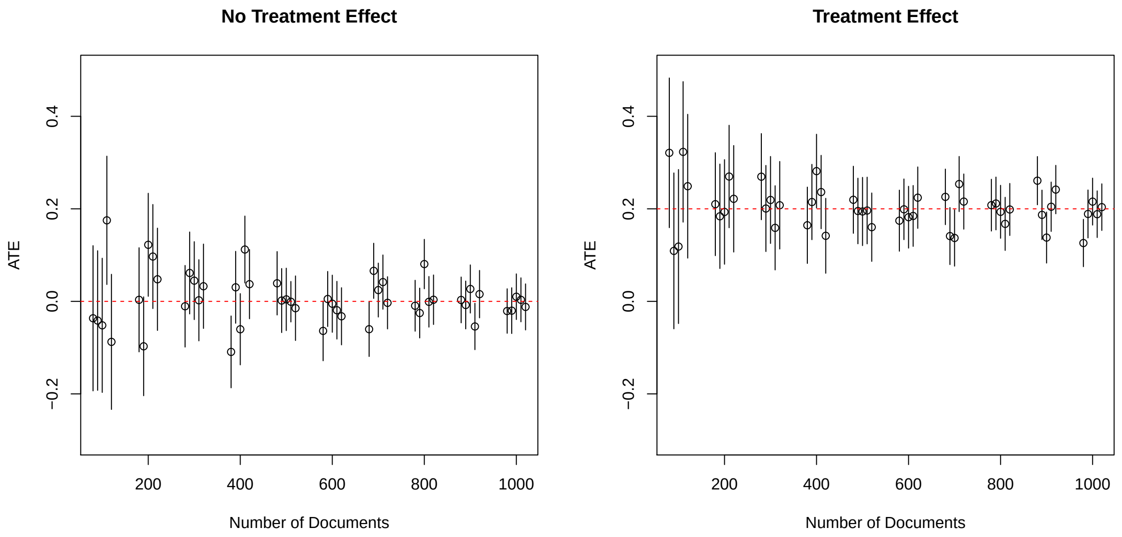


Figure 1: Estimated average treatment effect with 95% confidence intervals holding expected number of words per document fixed at 40, and the concentration parameter fixed at 1/3. The STM is able to recover the true ATE both in cases where there is no treatment effect (left) and cases with a sizable treatment effect (right). As expected, inferences improve as sample sizes increase.

$\zeta$ the expected number of words (40), the size of the ATE (0, .2), and $G_0$ the concentration parameter for the Dirichlet (1/3).

We estimate the STM model on each dataset using default parameters for the prior. We then calculate the modal estimates for the topic proportions ($\theta$) and perform a standard OLS regression to get estimates of the treatment effect and confidence intervals.

The results are shown in Figure 1. In the case of no treatment effect as well as a sizable treatment effect, the STM is able to recover the effects of interest. As expected, uncertainty shrinks with sample size.

**Interaction Simulations** Here we consider a slightly more complex simulation. Specifically we consider a continuous variable which under control has a negative effect on a topic, and under treatment has a positive effect on a topic. In each case we simulate with 100 documents, a vocabulary size of 500 and 50 words per document in expectation. The

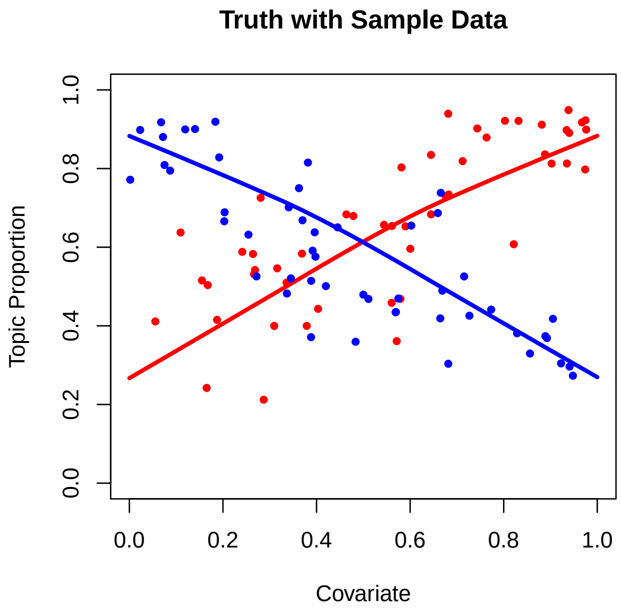


Figure 2: Simulated covariate effect, where treatment assignment causes a continuous variable along the $X$ axis to have either a positive or negative effect on a topic. Lines shown are the true relationships for each simulation while the dots indicate the true latent parameters for a single dataset.

true relationships are plotted in Figure 2 along with a sample of what the true latent data might look like for a particular simulation.

Using the STM and LDA we fit a model to each of 250 simulated datasets. For each dataset we plot a single line to indicate the treatment effect and the control effect (separated here for clarity). Lines are plotted semi-transparently for visibility with the true relationship super-imposed as a dotted black line. The fitted lines are shown in Figure 3.

Since it can be difficult to see the distribution, we show a histogram of the recovered slopes with a thicker black line demarcating the true value. These are plotted in Figure 4.

These simulations indicate that while LDA does sometimes correctly find the relationship it often is unable to capture the dynamic appropriately, in some cases reversing

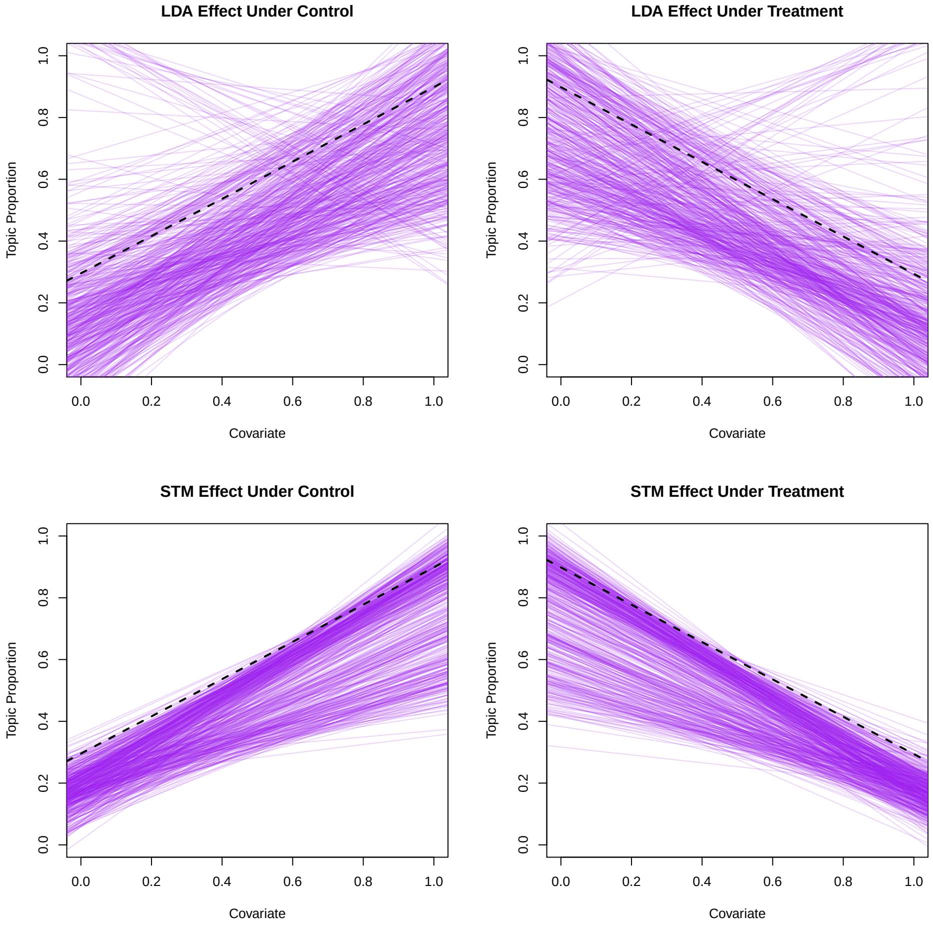


Figure 3: Covariate relationships recovered for treatment and control cases by LDA and STM.

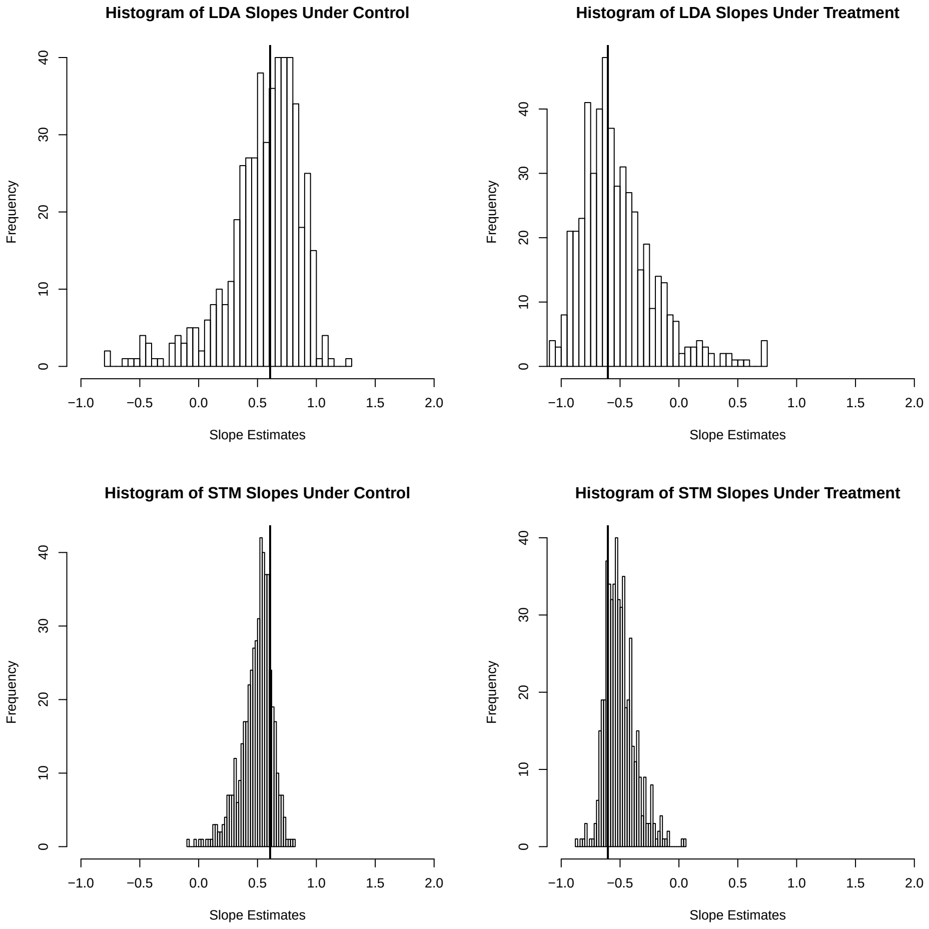


Figure 4: Slopes of continuous covariate effects recovered by STM and LDA

the sign of the effect. The STM by contrast tightly adheres to the true effect.

The reason the majority of the simulations appear below the line in both cases is an artifact of the data generating process. Document proportions are simulated in a continuous space but in practice the expected topic proportion is effectively discretized by the number of words. As a result, the true proportions are going to be slightly underestimated at both ends. The element of interest, both for the simulation as well as in practical examples is the slope of the line which constitutes the effect of the covariate under the two regimes.

### 1.2.6 Permutation Analysis

In this section we use one of our examples of real text data to show that when we randomly permute treatment assignment between text documents, we do not recover a treatment effect. This is further evidence that our model does not induce an effect when one does not in fact exist. In addition, under the true treatment assignment we see the most extreme effect, verifying that our results are not an artifact of the model, but instead reflect a consistent pattern within the data.

To do this, we use the Gadarian and Albertson (2013) data to estimate the effect of treatment on anxiety toward immigrants.[4] We estimate our model 100 times on this data, each time with a random permutation of treatment and control assignment to the documents. We then calculate the largest effect of treatment on any topic. If the results relating treatment to topics were an artifact of the model, then we would find a significant treatment effect regardless of how we assigned treatment to documents. If the results were a reflection of a true relationship between treatment and topics within the data, we would only find a treatment effect in the case where the assignment of treatment and control align with the true data.

Figure 5 shows the results of the permutation test. Most of the models have effect sizes clustered around zero, but the estimation that included the true assignment of treatment and control, indicated by the dotted line, is far to the right of zero. This indicates that the estimation itself is not producing the effect, but rather that the relationship between

[4] To simplify the permutation test, we do not include Party ID as a covariate.

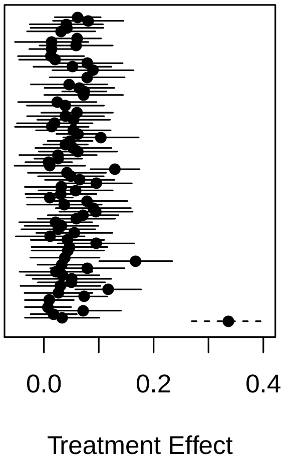


Figure 5: Maximum Treatment Effect Across Permutations

treatment and topics arises within the data.

Note that more than 5% of the intervals don't cover zero, this arises, among other factors, from searching for the largest effect size. Such a search would be inappropriate in the standard use of permutation tests but provides a more rigorous test of our claim. It also is suggestive evidence against a concern with multiple testing in this framework. We return to this issues later, but essentially were multiple testing over topics putting us at risk of finding consistently large spurious effects, we would also see large spurious effects in our permutation test. As we don't, we take that as evidence for a cautious optimism that multiple testing concerns are not a huge problem here.

### 1.2.7 Comparison with LDA

An important question we asked ourselves in developing our approach was 'is there any need to go beyond a two-step process using Latent Dirichlet Allocation (LDA) first and then estimating covariate relationships after the fact?' We addressed this with simulated data above, but it is useful to add to those findings an assessment on real data. Specifically we are interested in characterizing whether LDA is able to recover a simple binary treatment effect in one of our real datasets.

In order to demonstrate the differences in solutions attained by the two methods, we re-analyze the immigration data from Gadarian and Albertson (2013). As with STM, we first remove all stopwords and stem the corpus before conducting our analysis. We take an additional step of removing any word which only appears once in the corpus and any documents which subsequently contain no words. The unique words cause instability in the LDA analyses whereas STM is relatively unaffected due to the use of prior information.[5] We estimate LDA using collapsed Gibbs sampling (CGS), with fixed hyperparameters both at .05, running for 1000 iterations discarding 950 for burnin.[6] We estimate STM using default priors and using *only* the binary treatment indicator. As in the permutation test this simplifies the interpretation and crucially it biases *against* STM because it means we are using considerably less information than is available.

We start by running each algorithm 250 times with different starting values. We evaluate each run on our two selection criteria: *semantic coherence*, which measures the frequency with which high probability topic words tend to co-occur in documents, and *exclusivity* which measures the share of top topic words which are distinct to a given topic. In each case we use a comparison set size of 20 which was chosen in advance of viewing the results. We emphasize that our model does not directly optimize either criterion and thus

[5] The stabilizing influence is most likely a result of the structured priors in STM. Inference for topic assignment of words which only appear once comes from other words in the same document as the single-appearance words. Thus imagine the word “unlikely” appears only once in the corpus. In LDA its topic assignment will be based on the proportion of words assigned to each topic within the same document. In STM the assignment will be based both on the other words within the same document, but also other words in documents with similar covariate profiles. This tends to lead to inference which is more stable across initializations.

[6] Collapsed Gibbs sampling (CGS) for LDA is far in excess of what is needed for convergence. Many analysts simply use the final pass rather than averaging over iterations, but we use the last 50 in order to decrease sampling variability. To remove sampling variability as an issue we had originally used a variational approximation of LDA, but the algorithm produced pathological results for the K=3 case we consider here when estimating the Dirichlet shape parameter on topic prevalence. We do however emphasize that CGS is generally considered to be more accurate than standard variational approximation for LDA.

it provides a fair basis for comparison between the two models. Figure 6 shows the space of all LDA and STM solutions with larger numbers being better along each dimension.

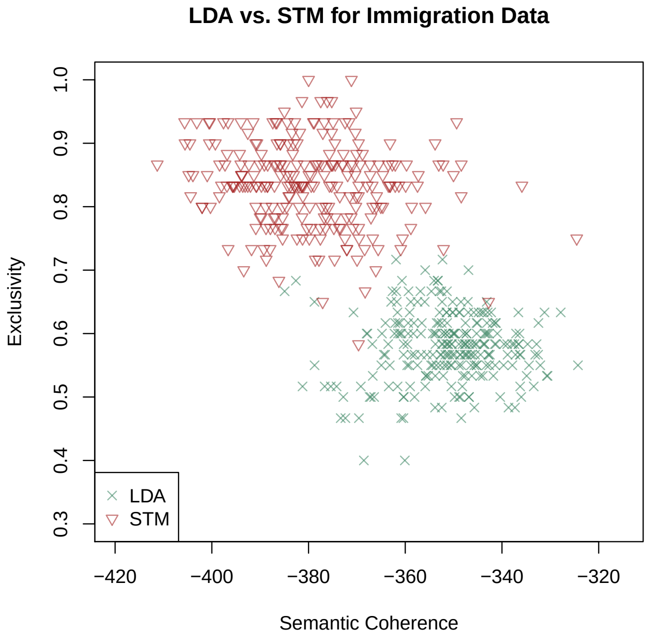


Figure 6: Semantic Coherence and Exclusivity for 250 runs of LDA and STM using different starting values.

We note that in general, STM provides more exclusive solutions where LDA favors semantic coherence. We note however that STM does provide a few solutions with the maximum level of semantic coherence attained by either model but with higher exclusivity than LDA. The plot also highlights the implicit tradeoff between coherence and exclusivity; that is, it is trivially easy to have either extremely high coherence (by having all topics have the same top words) or high exclusivity (by picking completely disjoint sets which do not co-occur in actual documents) and thus it is useful to examine both criteria in concert. We hope to explore whether these general trends hold across different document sets in future work.

In order to examine more closely examine results that are favorable to each model, we randomly choose a high performing solution using the same steps discussed in the paper. First we discard any solution below the 75% quantile along each dimension. Then we randomly select a solution from the remaining models for analysis. While in actual data analysis it would be best to look at several options and choose the most useful representation of the data, for the purposes of fairly comparing the two algorithms we made one random selection and did not search over other alternatives.

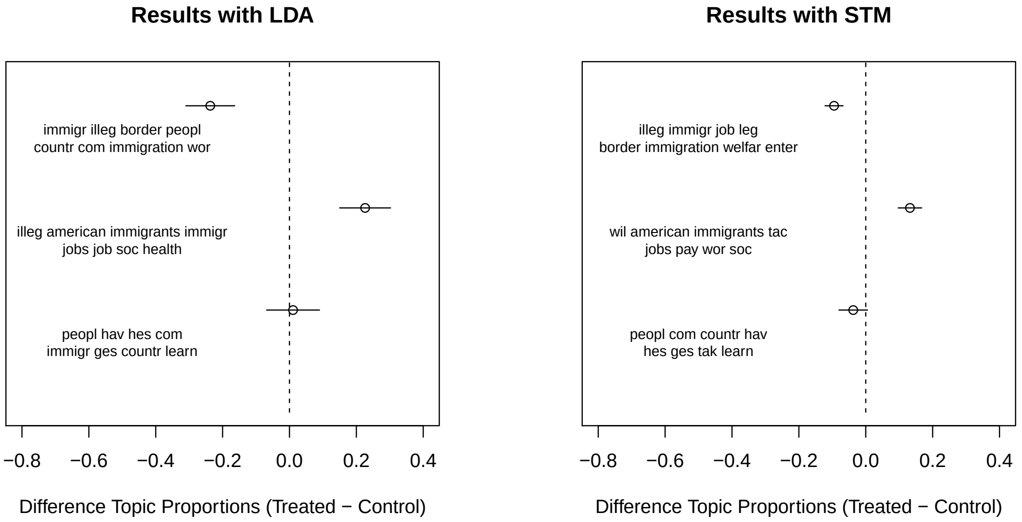


Figure 7: Treatment effects from a randomly selected high performing LDA and STM run on the immigration data.

We plot the two results in Figure 7 (aligning the results so comparable topics are on the same line). Words shown are the highest probability word stems (rather than the FREX words we use elsewhere in the paper, here we use highest probability words for comparability with standard LDA practice). Our model selection procedures are successful in that both solutions are reasonable and quite similar. The estimates of the treatment effects for LDA are higher but with much larger confidence intervals suggesting considerably more heterogeneity in the treatment and control groups.

This is consistent with the topics themselves which are more focused in the STM model. Looking over the different topics from the LDA estimation, we see that the words are not very exclusive to topics: “immigr”, “illeg”, “people”, “countr”, and “com” all appear in multiple topics. In contrast, the STM has much more exclusive topics: there are no overlapping words across topics. Even though the topics between LDA and STM are quite similar, interpretation in the more exclusive STM topics is more straightforward. The topic reflects both that the respondent is talking about immigration and the reasons for the respondent's concern. By contrast in the LDA version of the topic, the user can only determine the general topic, which seems to be common across all estimated topics.

This analysis is intended to provide a single concrete example to show the contrast between LDA and STM with real data. We emphasize that these results should not be taken as universal and analysis was done to emphasize comparability between the two methods as opposed to best practice. Thus we highlight two points: (a) LDA will not generally produce larger treatment effect estimates with wider confidence intervals, and (b) a careful analysis would examine documents rather than just looking at highest probability words. Our analysis of the immigration data is contained in the main body of the paper, but this section's analysis helps to contrast the two approaches with actual documents, as a complement to the simulated data used above.

### 1.2.8 Comparison with Hand-Coded Results

Here we cover the specifics of our comparisons between our results and the results from the hand-coding. The first thing to note is our topics do not match up perfectly with the categories that the coders used. This should not be a surprise, since we are using an unsupervised method whereas the human coders received supervision via a set of instructions. In particular, our topics do not have many words that seem to evoke "concern" or "enthusiasm" for immigrants. However, this is somewhat consistent with the human coded results because the human coders coded very few documents as expressing these topics. RA 1 classified only 18% of those who took the survey as having either enthusiasm or concern for immigrants. RA 2 classified 23% as having either enthusiasm or concern. These low numbers are in line with our analysis in that we did not retrieve very many words related to concern or enthusiasm and hence our results are consistent with Gadarian and Albertson (2013)

Even though the topics and coder categories do not match perfectly, the vocabulary

|  | No Concern | Concern |
|---|---|---|
| 0. Think | 0.76 | 0.23 |
| 1. Worried | 0.87 | 0.13 |

Table 2: Treatment and Concern for Immigrants, RA 1

|  | No Concern | Concern |
|---|---|---|
| 0. Think | 0.65 | 0.27 |
| 1. Worried | 0.76 | 0.16 |

Table 3: Treatment and Concern for Immigrants, RA 2

associated with Topic 1 is more closely in line with fear of immigrants and anger toward immigrants, and so we will compare Topic 1 to the fear and anger categories. Aggregating across the treatment conditions, RA1 classified 56% of respondents as having negative views of immigrants (either fear or anger), and RA2 classified 79% of respondents as having a negative view of immigrants. The data clearly has a much richer and present vocabulary that corresponds to fear and anger toward immigrants.

The treatment effect we uncovered using the Structural Topic Model corresponds to what Gadarian and Albertson (2013) find with human coders. To see this effect from the human coding, Table 1.2.8 and 1.2.8 shows the breakdown by treatment and control, where "Concern" refers to responses that were coded either into concern or enthusiasm. Tables 1.2.8 and 1.2.8 show the breakdown when looking at the negative views of immigrants that is captured by our Topic 1, where "Negative" refers to responses that were coded as either containing fear or containing anger toward immigrants. The treatment increases the likelihood of a response with fear or anger under the coding scheme, in line with what the topic model uncovers.

An additional way to compare our results with that of the human coders is to consider the correlation between the hand-coding and the document-level probabilities of topics.

|  | Not Negative | Negative |
|---|---|---|
| 0. Think | 0.60 | 0.39 |
| 1. Worried | 0.26 | 0.74 |

Table 4: Treatment and Negative Views Toward Immigrants, RA 1

|  | Not Negative | Negative |
|---|---|---|
| 0. Think | 0.32 | 0.60 |
| 1. Worried | 0.06 | 0.86 |

Table 5: Treatment and Negative Views Toward Immigrants, RA 2

Because the topics are different than the pre-determined hand-coded topics, we should not necessarily expect the predicted document proportions and the coders' classification to completely line up. This is particularly the case because the hand coders put the majority of posts into either fear or anger. However, we should see some relationship between the probability that a response is in the fear and anger topic estimated by the Structural Topic Model and the human coding of these same responses.

To examine document-level agreement, we take observations where the coders were in 100% agreement about the classification. We would expect documents where the coders agree to be the most clearly classifiable. Figure 8 presents a histogram of the documents by the predicted proportion of a document in that topic, coloring each document by the category given by the coders. The x-axis on the histogram is the predicted proportion of a document in Topic 1. Documents on the left side of the barplot have very low proportions of Topic 1, while on the right side of the histogram have very high proportions of Topic 1. The colors on the bar plot represent the coders' classification of the same documents. For example, all of the documents with between 0 and .1 proportion of Topic 1 are coded either as having enthusiasm or not given a category. On the other hand, almost all of the documents with over .8 of Topic 1 were either coded with fear or anger.

We do see a correlation between the topics assigned to documents by the topic model and the classification done by the coders. Documents with high proportions of Topic 1 are more likely to be coded with fear and anger, and are rarely coded with enthusiasm or not categorized. We might be able to, for example, use the topic probabilities generated by the topic model to predict whether the coders would categorize a post into either enthusiasm or no category with a fairly high success rate.

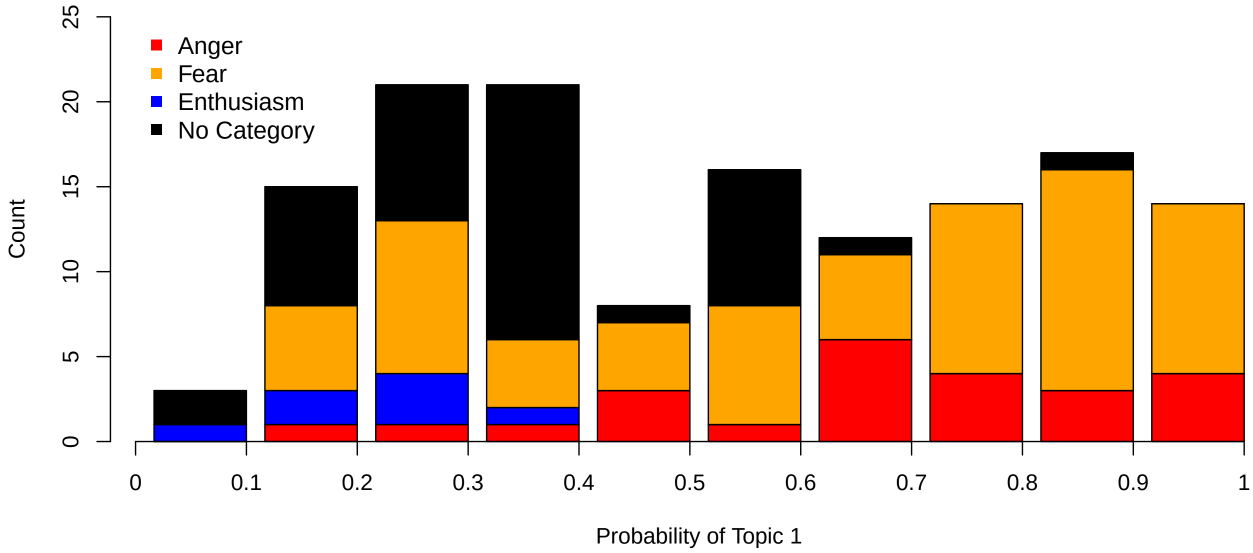


Figure 8: Coders' Classification Compared to Unsupervised Topic Proportions

## 1.3 Getting Started

In this section we discuss topics of interest to the applied user. Section 1.3.2 covers concerns about multiple testing as well as the application of pre-registration and false discovery rate methods. In Section 1.3.1, we describe the standard pre-processing algorithms we employ as well as introduce a new software tool for pre-processing and working with texts. In Section 1.3.3 we introduce a second software tool which provides an interactive visualization interface for exploring the results of topic models. In Section 1.3.4 we show how to use our approach to perform mediation analysis.

### 1.3.1 Text Preparation and Custom Pre-Processing Software

In our examples below, we adopt a set of procedures which are standard pre-processing steps in computational linguistics (Manning *et al.*, 2008). First we remove punctuation, capitalization, and stop words from the open-ended responses. Stop words are terms that are so common as to be uninformative. For example, the word "and" occurs in nearly all responses. As such, a respondent's use of the word "and" conveys little to no information about the topical content of the response. If stop words like "and" are not removed, the model may discover topics that are described primarily by stop words, due to their relatively high frequency of occurrence. And even if stop words end up scattered across topics, these associations are likely spurious.

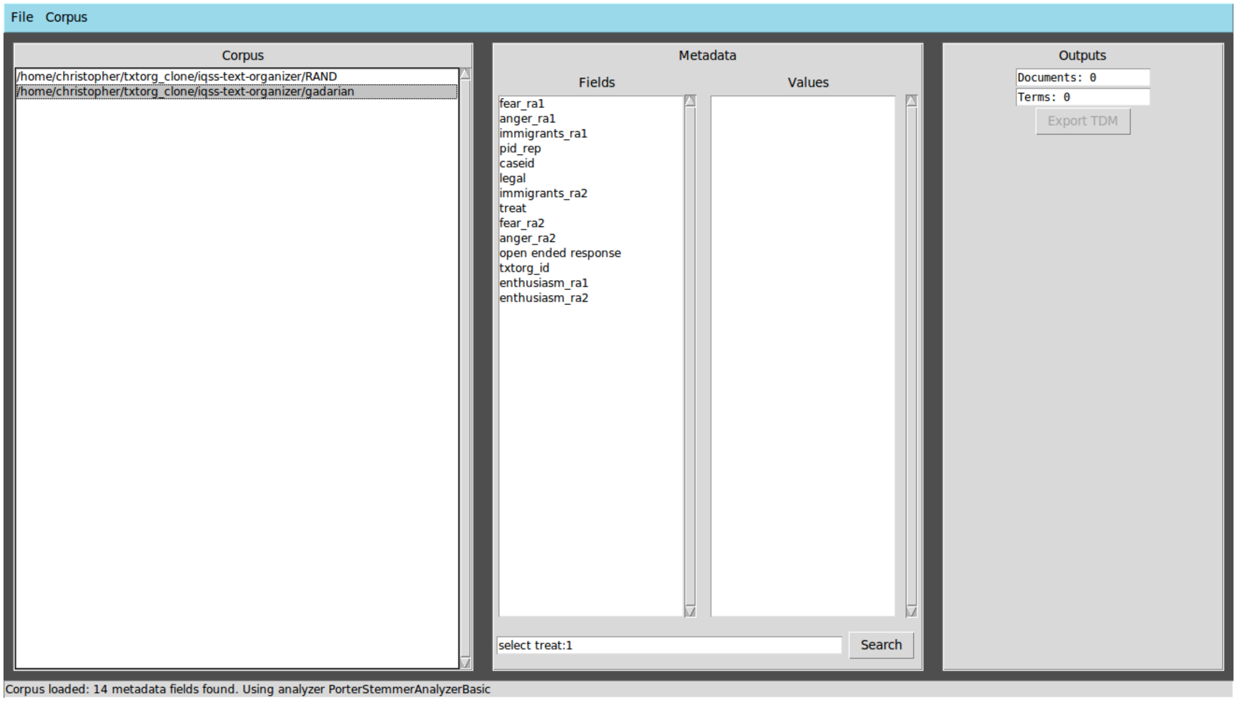
Next, the remaining words are *stemmed*. Stemming is the process of reducing inflected words to their lexical roots, or "stems." All the terms that share a common root are folded into a single term. This term need only be unique from all others; it does not need to be a real word. For example, in some implementations (including the one we use), the words happiness, happiest, and happier are all reduced to the stem "happi." Because the inflected variants of a common stem are probably all associated with a common topic, stemming generally improves performance with small data sets. Gains from stemming decrease with larger data sets, as the the model is less likely to spuriously assign variants of a common stem to different topics. In our analysis, we employ the canonical Porter Stemming Algorithm (Porter, 1980), which operates by algorithmically stripping away the suffixes of words and returning only the roots, or stems, of the original words.

The remaining terms are then transformed into a vector-space representation called a term-document matrix (tdm). For example, let $T$ index the unique terms in the data and $D$ index the documents. Let $M$ be a tdm. Then $M = [m_{t,d}]_{T \times D}$, where cell $m_{t,d}$ is the number of times term $t$ occurs in document $d$.[7] Column $m_{*,d}$ is then a vector representation of document $d$, which can be analyzed with standard techniques from multivariate analysis. All statistical analyses are computed on the tdm. To facilitate we wrote a multi-featured `txtorg` text management interface discussed below.

Notably, by discarding the structure and linear ordering of the words in the documents, the tdm makes the simplifying assumption that documents can be represented as an unordered "bag of words." While perhaps counter-intuitive, this approach yields a reasonable approximation while greatly improving computational efficiency.[8] We empha-

[7] Occasionally, cell values are indicators for whether or not term $t$ appeared in document $d$.

[8] Typically we use the space of single words, called unigrams. We can easily incorporate a level of word order by incorporating bi-grams or tri-grams. Unfortunately the number of terms grows extremely quickly. The consensus in the literature is that unigrams are sufficient for most tasks (Hopkins and King,



/home/christopher/txtorg_clone/iqss-text-organizer/RAND
/home/christopher/txtorg_clone/iqss-text-organizer/gadarian

fear_ra1
anger_ra1
immigrants_ra1
pid_rep
caseid
legal
immigrants_ra2
treat
fear_ra2
anger_ra2
open ended response
txtorg_id
enthusiasm_ra1
enthusiasm_ra2

select treat:1

```
Documents: 0
Terms: 0
```

Corpus loaded: 14 metadata fields found. Using analyzer PorterStemmerAnalyzerBasic

Figure 9: Screenshot of txtorg

size that these data preprocessing choices are separate from our method in the sense that the analyst can choose to apply them or not based on their particular case.

To facilitate the proposed workflow we developed a software package `Txtorg` (Lucas *et al.*, 2013) to prepare our data for analysis. While the **stm** package contains a a function `textProcessor` to setup data that is in a spreadsheet format, and can be processed by the **tm** package, `Txtorg` contains numerous other functionalities. `Txtorg` is a scalable Python-based tool with a full-featured graphical user interface that makes the management of textual data simple and intuitive for everyone. By leveraging the power of Apache Lucene (The Apache Software Foundation, 2013), `Txtorg` enables users to store and search large amounts of textual data that would otherwise be unwieldy. A screenshot of the `Txtorg` user interface is presented in Figure 9.

`Txtorg` uses Apache Lucene to create a search index, which allows for local storage and quick searching, even with large corpora. Search indices, like those created by `Txtorg`,

2010)

allow queries to return the relevant documents without scanning the entire corpus. Moreover, the index created by `Txtorg` is compressed to a fraction of the size of the original corpus, and from it, full corpora can be recovered. `Txtorg` also provides the user with a number of different preprocessing utilities, with which corpora can be cleaned prior to the indexing stage. And because `Txtorg` uses Apache Lucene, it can leverage a wide array of use cases, including non-English text and various export formats.

A typical workflow with `Txtorg` might then proceed as follows. Steps 4 - 7 may be repeated infinitely without repeating earlier steps.

1. **Collect data:** Here, `Txtorg` is agnostic, as the program can be used with a wide range of data types, ranging from survey responses to Chinese poetry written in Mandarin.

2. **Organize data:** Depending on the original source from which the data were collected, reformatting may be necessary. Specifically, `Txtorg` can import corpora stored in one of three formats - a field (column) in a csv file (common for data generated by a survey), a directory containing the individual documents as `.txt` files, or a csv with a field containing the file paths to the individual documents in the corpus. If the data are imported as a field in a csv file, the remaining columns in the csv are imported as metadata. For example, a researcher might upload responses to an open-ended survey question stored as a field in a csv. The remaining columns may be responses to closed-ended questions, such as sex, age, and education. If the data are not imported as a field in a csv, metadata may be imported separated as a csv.

3. **Clean the data:** Through Python, we provide several important preprocessing utilities within `Txtorg`. For example, `Txtorg` can convert numerous encoding schemes to UTF-8 and replace user-specified strings of text with alternative text strings.

4. **Import the data:** Select the import data option from a drop-down menu. The corpus is then indexed and the index is saved locally.

5. **Preprocessing:** Next, the user may choose to implement any of several forms of preprocessing supported by `Txtorg`. Most importantly, `Txtorg` supports stemming, stop word removal, and phrase indexing. `Txtorg` implements the canonical Porter stemming algorithm, which users may simply turn on and off at any time. Similarly, users can turn stop word removal on and off as they please, and may change the list of stop words at any time. Phrase indexing is somewhat less flexible. By default, the `Txtorg` indexes all single-word tokens (unigrams). If a researcher wishes to search and index tokens longer than a single gram (for example, "border security"), these grams must be imported at step 3, when the user imports the data. If after importing the data, the user wishes to index phrases longer than a single gram, the corpus may be re-indexed with an updated list of phrases.

6. **Searching the corpus:** Next, the user may search and subset the index by the terms in the individual documents or by document metadata. The search function is sophisticated, allowing users to string together multiple searches. For example, a researcher might be interested in the set of documents written by individuals in specific treatment conditions, or by men between the ages of 18 and 30 who are either Hispanic or black mentioning "border security" or "national security." `Txtorg` can implement searches of this form and complexity, as well as more simple searches. Finally, users may also select the entire corpus.

7. **Exporting the data:** After selecting the documents of interest, users can export the data as either a term-document matrix or as entire documents. `Txtorg` supports two primary TDM formats, the LDA sparse matrix and the standard, flat csv. Documents exported in their entirety are written to a directory as txt files. Users can also restrict the terms that appear in the TDM according to frequency of occurence.

8. **Analysis:** Finally, the user may conduct any sort of analysis on the exported data, including those based on the structural topic model explained in this paper, but also analyses of other forms, from LDA to manually reading the documents in their unprocessed entirety.

Perhaps Txtorg's most compelling feature is its speed, especially with large corpora. Sequentially searching through all responses to subset on a specific term is incredibly slow, and literally impossible in some cases. By creating an index, search time decreases by orders of magnitude. One can easily analyze just those responses containing "immigration reform," then seconds later export and analyze the TDM for responses containing "border security".

A second selling point is the program's ability to subset and sort data without editing the original index. Perhaps a researcher conducts an initial analysis on responses from college students, then in a subsequent project, wishes to examine racial minorities. Or one might want to examine the responses mentioning "President" separately from the full corpus. Subsetting responses with a simple script would be tedious in cases where the data are small, and impossible where they are large. With Txtorg, after the corpus is indexed, researchers can quickly pull from it as many unique TDMs as they like, at any point in time.

### 1.3.2 Multiple Testing

There is reasonable concern that with large groups of topics multiple testing will cause the researcher to find spurious relationships between covariates and topics. Put intuitively the idea can be stated that as the number of topics grows, iteratively testing treatment effects over all topics in the model will cause a false positive significant result. We discuss why this concern might not be as troubling as it first appears, and then discuss how certain methods can be used to address it.

On one view, standard p-values are misleading in this setting because the null distribution is not particularly believable (Gelman *et al.*, 2012). That is, we chose to incorporate a covariate into the model because we believe that there is an effect on word choice and it is unlikely that any effect would be exactly zero. Thus the question is really more about estimating effect sizes correctly rather than rejections of a sharp null. While this view is appealing, it still does not address the nagging concern that comparison over many topics will tend to cause us to see spurious relationships where there are none.

Even if there is a spurious relationship found by the model, it does not mean it is a substantively interesting topic. For the researcher to come to an incorrect conclusion there must be a significant covariate on a topic that has a semantically coherent and intellectually relevant meaning. For those who still have concerns, we conducted a permutation test in the previous section. There we showed that even when searching for the maximum effect in the permuted data, the true effect with the real treatment indicator was an enormous outlier. This suggests that spurious effects are not as common as people might fear.

Still for those willing to expend additional planning and effort, it is possible to further hedge against the risk of false positives. In some experimental settings, researchers limit their risk by adopting pre-analysis plans. These plans specify which relationships they will test and what they will expect to find. In the topic model setting, researchers can specify word groups that they expect to co-occur and the effect they would expect to find on such topics. This could be as informal as expecting to find that issues about the 'economy' are likely to appear in the ANES most important problems and the likely relationship with household income. On the more rigorous side, analysts could specify a list of word stems beforehand and only consider topics which contain a certain percentage of those key words.

An alternative approach is to condition on the model results but augment testing with False Discovery Rate methods (see Efron (2010) for an overview). Conditioning on the model we treat the topic proportions as observed data and apply standard corrections as with any other dataset. These methods have been shown to be effective in genetics where there are often tens of thousands of tests; by contrast, we will typically have far fewer topics.

### 1.3.3 Post-analysis visualization

For those unfamiliar with topic modeling the output from standard software can be overwhelming. Models typically contain thousands of parameters which must be summarized for user interpretation. Typically an analyst will develop custom functions which summarize parameters of interest for their specific application. In addition to a rich set of functions for performing the types of visualizations reported in this paper, we have also been developing a more general visualization and model summary tool which builds on recent efforts in computer science (Gardner *et al.*, 2010; Chaney and Blei, 2012).

Here we highlight our ongoing work on the Structural Topic Model Browser (STMB). The STMB visualizes output from the model in a standard web browsing format. It incorporates information at the corpus level as well as individual views of each document. We believe that browsers of this sort serve a crucial role in connecting the end user to the original text, and help to make good on the promise of topic model as a method for *computer-assisted reading.* The software will be made available as an open-source project on Github which will allow easy access for the end user as well as other developers who may wish to adapt the code or add features.

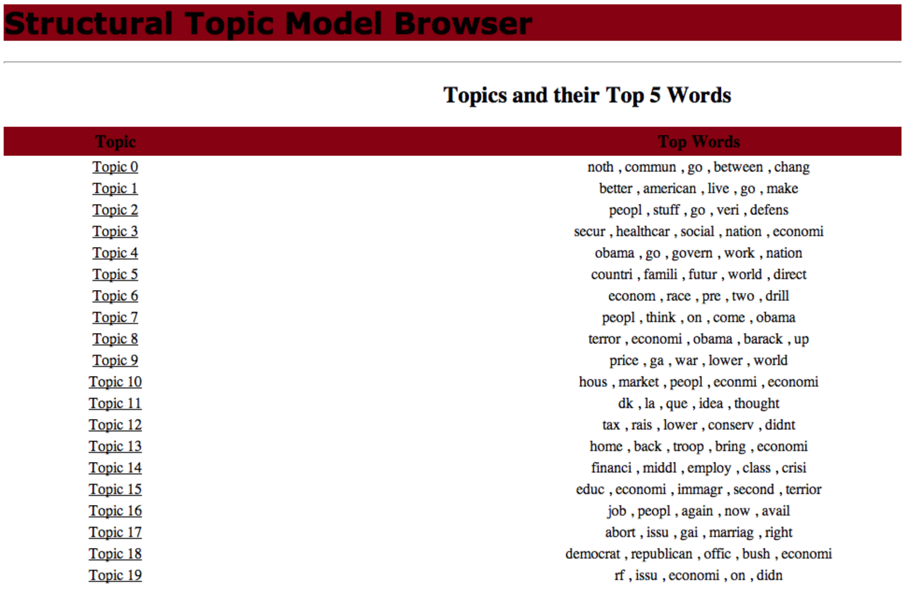
To use the browser, the user passes the original documents and covariate information as well as the model output to a visualization script written in Python. In addition, the user specifies the number of words to be displayed for a topic. In this example, we're exploring a structural topic model of the ANES data that combined all of the most important problem questions with 55 topics and 5 words displayed per topic.

The Index Page for the website lists each topic and the most probable words under that topic. Figure 10 shows the Index Page of the browser for the ANES dataset.

From the browser page, each of the topic listings, e.g. "Topic 10", function as links to individual Topic Pages. The Topic Page contains a number of important metrics about the topic's fit and relationship to documents. In addition to the top words as displayed on the index page, it contains the top FREX-scored words, as described in Section 1.1.4.

The Topic Page contains the topic's expected frequency across a corpus, which is the mean proportion of words across the documents that are assigned to this topic. Furthermore, the Topic Page displays the topic's semantic coherence score, a measure of the topic's fit within documents, where larger numbers indicate a better fit (Mimno *et al.*, 2011).

Below the expected frequency and semantic coherence scores, the browser displays the top words in context as in Gardner *et al.* (2010). For each top word, there is a sample from one of the documents with a high proportion of words assigned to the specific topic

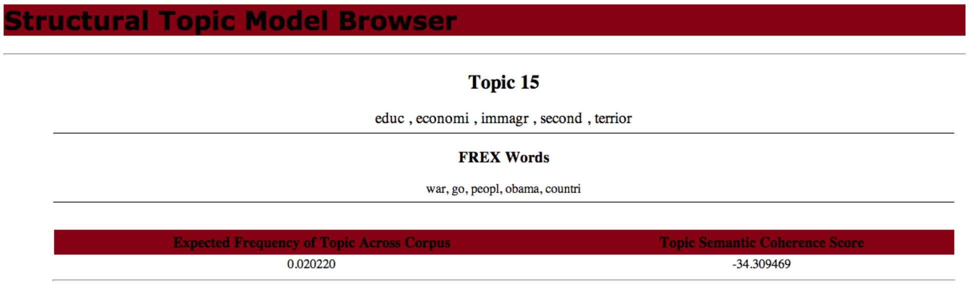


Structural Topic Model Browser

Topics and their Top 5 Words

| Topic | Top Words |
|---|---|
| Topic 0 | noth , commun , go , between , chang |
| Topic 1 | better , american , live , go , make |
| Topic 2 | peopl , stuff , go , veri , defens |
| Topic 3 | secur , healthcar , social , nation , economi |
| Topic 4 | obama , go , govern , work , nation |
| Topic 5 | countri , famili , futur , world , direct |
| Topic 6 | econom , race , pre , two , drill |
| Topic 7 | peopl , think , on , come , obama |
| Topic 8 | terror , economi , obama , barack , up |
| Topic 9 | price , ga , war , lower , world |
| Topic 10 | hous , market , peopl , econmi , economi |
| Topic 11 | dk , la , que , idea , thought |
| Topic 12 | tax , rais , lower , conserv , didnt |
| Topic 13 | home , back , troop , bring , economi |
| Topic 14 | financi , middl , employ , class , crisi |
| Topic 15 | educ , economi , immagr , second , terrior |
| Topic 16 | job , peopl , again , now , avail |
| Topic 17 | abort , issu , gai , marriag , right |
| Topic 18 | democrat , republican , offic , bush , economi |
| Topic 19 | rf , issu , economi , on , didn |

Figure 10: ANES Structural Topic Model Browser Index Page



Structural Topic Model Browser

Topic 15

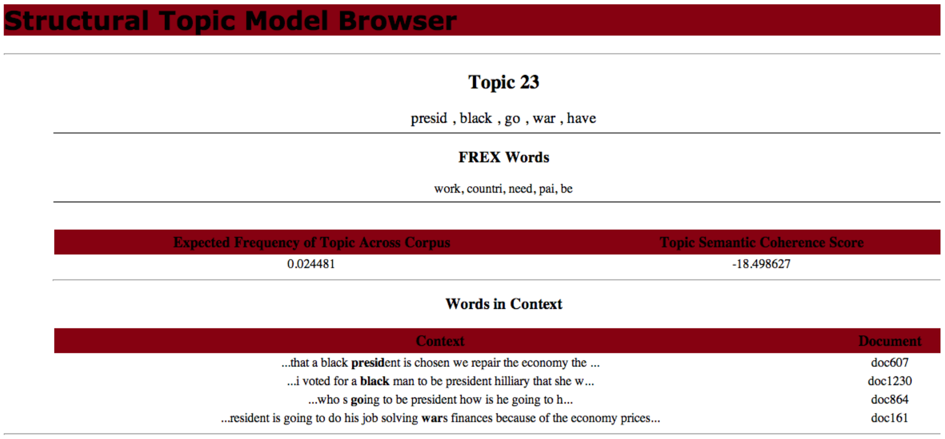
educ , economi , immagr , second , terrior

FREX Words

war, go, peopl, obama, countri

| Expected Frequency of Topic Across Corpus | Topic Semantic Coherence Score |
|---|---|
| 0.020220 | -34.309469 |

Figure 11: Topic Page for Topic 15



Structural Topic Model Browser

**Topic 23**

presid , black , go , war , have

**FREX Words**

work, countri, need, pai, be

| Expected Frequency of Topic Across Corpus | Topic Semantic Coherence Score |
|---|---|
| 0.024481 | -18.498627 |

**Words in Context**

| Context | Document |
|---|---|
| ...that a black **presid**ent is chosen we repair the economy the ... | doc607 |
| ...i voted for a **black** man to be president hilliary that she w... | doc1230 |
| ...who s **go**ing to be president how is he going to h... | doc864 |
| ...resident is going to do his job solving **war**s finances because of the economy prices... | doc161 |

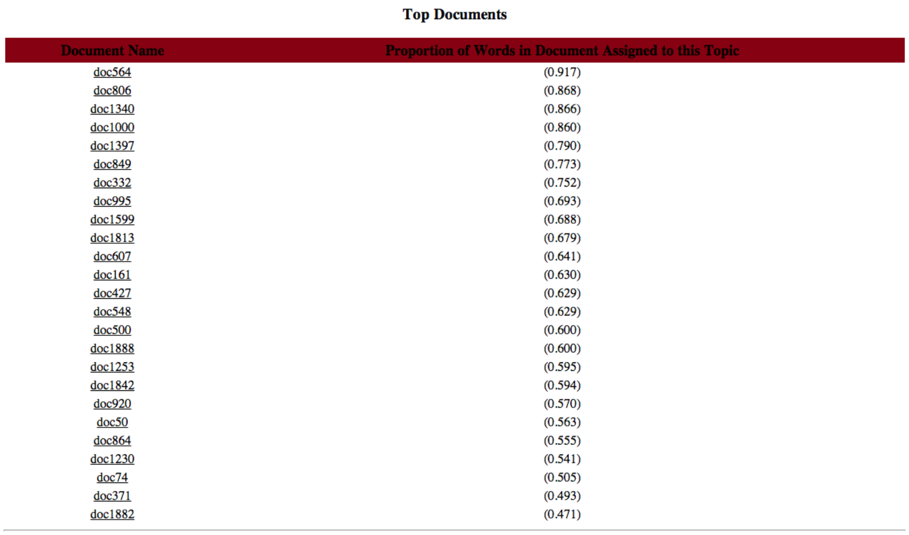
Figure 12: Top of the Topic Page for Topic 23

and with high incidence of the specific word. This sample presents the word in bold, with forty characters to each side, and the document listed to the right. This allows the user to connect the top words in the topic to an instance in which this has been used. Figure 12 shows the top of the Topic Page for topic 23, including the context section. The topic's top two words are 'president' and 'black', and we see in the context two documents describing how the participants specifically voted for a black president.

After the words are presented in context, we present the top 25 documents by the proportion of words assigned to this topic. Each document name is also a link to that document's page. This allows the user to see roughly the level at which documents that use this document heavily pick words from this topic, as well as providing links to the individual document pages, described below. Figure 13 shows the 'Top Documents' section of the Topic Page for topic 23.

The last element on the topic page Topic Page is a set of graphs of the documents' expected topic proportions plotted against the covariate values for that document. This allows the user to visualize how the covariate impacts the topic's fit to individual document and to spot any patterns for further analysis. Figures 14 show the covariate plot for topic 23 with respect to the Age.

Each document has its own page, linked to by the Topic Pages. The Document Page contains a list of the Top 10 topics associated with this document, the proportion of words



**Top Documents**

| Document Name | Proportion of Words in Document Assigned to this Topic |
|---|---|
| doc564 | (0.917) |
| doc806 | (0.868) |
| doc1340 | (0.866) |
| doc1000 | (0.860) |
| doc1397 | (0.790) |
| doc849 | (0.773) |
| doc332 | (0.752) |
| doc995 | (0.693) |
| doc1599 | (0.688) |
| doc1813 | (0.679) |
| doc607 | (0.641) |
| doc161 | (0.630) |
| doc427 | (0.629) |
| doc548 | (0.629) |
| doc500 | (0.600) |
| doc1888 | (0.600) |
| doc1253 | (0.595) |
| doc1842 | (0.594) |
| doc920 | (0.570) |
| doc50 | (0.563) |
| doc864 | (0.555) |
| doc1230 | (0.541) |
| doc74 | (0.505) |
| doc371 | (0.493) |
| doc1882 | (0.471) |

Figure 13: Top Documents for Topic 23

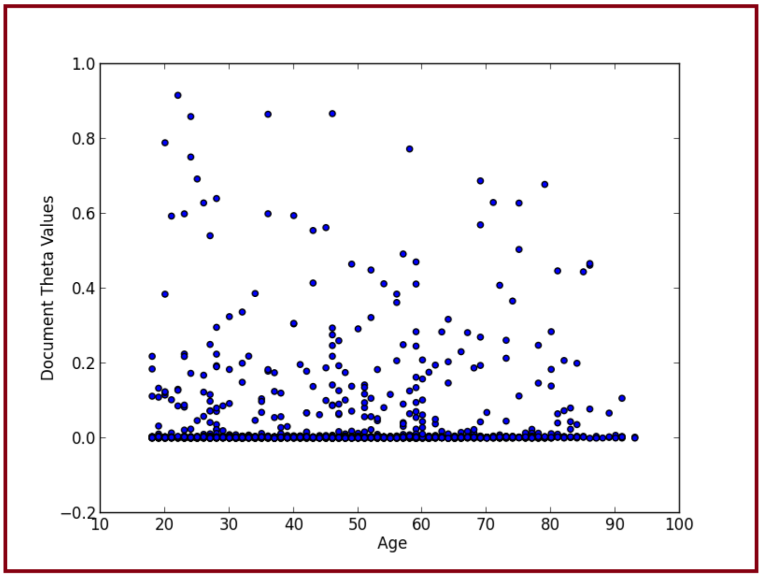
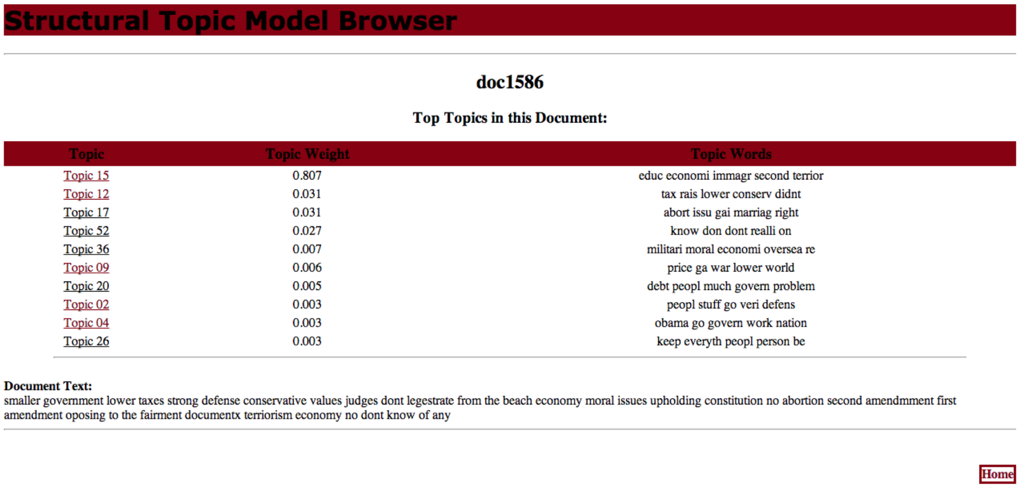


Figure 14: Age Covariate, Topic 23



Structural Topic Model Browser

doc1586

Top Topics in this Document:

| Topic | Topic Weight | Topic Words |
|---|---|---|
| Topic 15 | 0.807 | educ economi immagr second terrior |
| Topic 12 | 0.031 | tax rais lower conserv didnt |
| Topic 17 | 0.031 | abort issu gai marriag right |
| Topic 52 | 0.027 | know don dont realli on |
| Topic 36 | 0.007 | militari moral economi oversea re |
| Topic 09 | 0.006 | price ga war lower world |
| Topic 20 | 0.005 | debt peopl much govern problem |
| Topic 02 | 0.003 | peopl stuff go veri defens |
| Topic 04 | 0.003 | obama go govern work nation |
| Topic 26 | 0.003 | keep everyth peopl person be |

**Document Text:**
smaller government lower taxes strong defense conservative values judges dont legestrate from the beach economy moral issues upholding constitution no abortion second amendmment first amendment oposing to the fairment documentx terriorism economy no dont know of any

Figure 15: Document 1586

in the document assigned to that topic, and the topic $n$ words in that topic. Below, the browser displays the entire text of the document. Figure 15 is an example of a Document Page from the ANES topic model. Document 1586 scores highly in Topic 15, described above. We see from the text and topic fit, particularly the use of topic 12, that this is an example of a conservative response to the ANES questions.

Overall, the Structural Topic Model Browser provides a way for the user to use the output of the topic model to explore the relationship between text and the covariates. By mixing graphical representation, samples from the original texts, and quantitative measures of fit, the browser facilitates broader understanding of the analysis and intuitive ideas about patterns observable in the text.

### 1.3.4 Mediation Analysis

The output of the structural topic model can easily be combined with standard analysis techniques. In this section we show how this can be done with mediation analysis, though we stress that this application does not jointly estimate all parameters which would be useful future work. In this application we examine how ex-post strategy choice descriptions mediate the relationship between Rand *et al.* (2012)'s experimental intervention of encouraging intuitiveness or deliberation. In particular, for every subject we know their treatment assignment, their explanation of strategy choice, and their contribution. We can then take their explanation of their strategy choice and analyze it using the STM, utilizing the information we know about their treatment assignment. Then, for each subject, we have an estimate of $\theta$ for each of the topics (proportion of words from that topic). We can then apply standard mediation analysis using this $\theta$ as the mediator.

To conduct the mediation analysis we utilize the `mediation` package in R. This entails fitting two parametric models. The first is a regression of the topic representing intuitive thinking on the treatment condition. The second is a regression of normalized contributions $(0, 1)$ on both the treatment and the intuitive topic proportion. We expect the mediation effect to be positive, with the treatment positive impacting the intuition topic, and the intuition topic positively influencing contributions. Use of a tobit model for the contributions equation produced similar results.

Table 16 presents the average causal mediation effect along with 95% confidence intervals. We observe a positive mediation effect of approximately .09. An alternative form of analysis could use the open ended text that respondents wrote when first responding to the encouragement to write about an intuitive topic. This would avoid the problem of using ex-post rationalizations as a mediating variable.

## References

Bischof, J. and Airoldi, E. (2012). Summarizing topical content with word frequency and exclusivity. *arXiv preprint arXiv:1206.4631*.

Bishop, C. (2007). *Pattern Recognition and Machine Learning*. Springer, Cambridge,

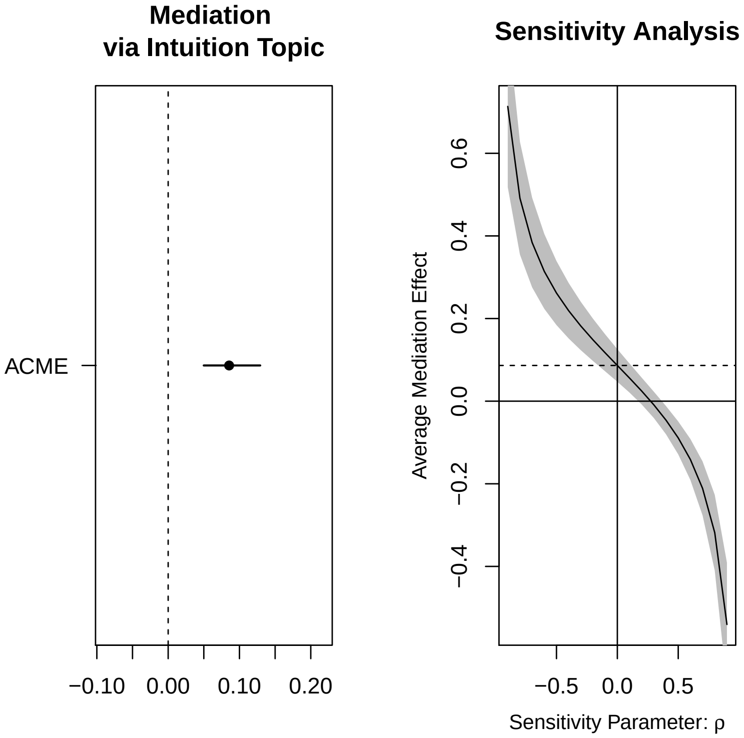


Figure 16: Left side of presents the average causal mediation effect (ACME) of intuition topic on normalized contributions using intuition encouragement design. Right side presents a formal sensitivity analysis as described in Imai *et al.* (2011).

MA.

Blackwell, M., Honaker, J., and King, G. (2011). Multiple overimputation: A unified approach to measurement error and missing data.

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

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

Buntine, W. L. and Jakulin, A. (2005). Discrete component analysis. In *SLSFS*, pages 1–33.

Chaney, A. and Blei, D. (2012). Visualizing topic models. *Department of Computer Science, Princeton University, Princeton, NJ, USA*.

Efron, B. (2010). *Large-scale inference: empirical Bayes methods for estimation, testing, and prediction*, volume 1. Cambridge University Press.

Eisenstein, J., Ahmed, A., and Xing, E. P. (2011). Sparse additive generative models of text. In *Proceedings of ICML*, pages 1041–1048.

Gadarian, S. and Albertson, B. (2013). Anxiety, immigration, and the search for information. *Political Psychology*.

Gardner, M., Lutes, J., Lund, J., Hansen, J., Walker, D., Ringger, E., and Seppi, K. (2010). The topic browser: An interactive tool for browsing topic models. In *NIPS Workshop on Challenges of Data Visualization. MIT Press*.

Gelman, A. and Hill, J. (2007). *Data analysis using regression and multilevel/hierarchical models*, volume 3. Cambridge University Press New York.

Gelman, A., Hill, J., and Yajima, M. (2012). Why we (usually) don't have to worry about multiple comparisons. *Journal of Research on Educational Effectiveness*, **5**(2), 189–211.

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

Grimmer, J. (2011). An introduction to bayesian inference via variational approximations. *Political Analysis*, **19**(1), 32–47.

Grimmer, J. and Stewart, B. M. (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, **21**(2).

Hopkins, D. (2012). The exaggerated life of death panels: The limits of framing effects on health care attitudes.

Hopkins, D. and King, G. (2010). A method of automated nonparametric content analysis for social science. *American Journal of Political Science*, **54**(1), 229–247. http://gking.harvard.edu/files/abs/words-abs.shtml.

Hornik, K. (2005). A CLUE for CLUster Ensembles. *Journal of Statistical Software*, **14**(12).

Imai, K., Keele, L., Tingley, D., and Yamamoto, T. (2011). Unpacking the black box of causality: Learning about causal mechanisms from experimental and observational studies. *American Political Science Review*, **105**(4), 765–789.

Krippendorff, K. (2004). *Content Analysis: An Introduction to Its Methodology*. Sage, New York.

Laver, M., Benoit, K., and Garry, J. (2003). Extracting policy positions from political texts using words as data. *American Political Science Review*, **97**(2), 311–331.

Lucas, C., Storer, A., and Tingley, D. (2013). Txtorg: A lucene-based software package for organizing textual data.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *An Introduction to Information Retrieval*. Cambridge University Press, Cambridge, England.

Mimno, D. and Blei, D. (2011). Bayesian checking for topic models. *Empirical Methods in Natural Language Processing*.

Mimno, D., Wallach, H. M., Talley, E., Leenders, M., and McCallum, A. (2011). Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 262–272. Association for Computational Linguistics.

Papadimitriou, C. H. and Steiglitz, K. (1998). *Combinatorial optimization: algorithms and complexity*. Courier Dover Publications.

Porter, M. (1980). An algorithm for suffix stripping. *Program*, **14**(3), 130–137.

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

Rand, D. G., Greene, J. D., and Nowak, M. A. (2012). Spontaneous giving and calculated greed. *Nature*, **489**(7416), 427–430.

Simon, A. and Xenos, M. (2004). Dimensional reduction of word-frequency data as a substitute for intersubjective content analysis. *Political Analysis*, **12**(1), 63–75.

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

Sontag, D. and Roy, D. (2009). Complexity of inference in topic models. In *NIPS Workshop on Applications for Topic Models: Text and Beyond*. Citeseer.

Taddy, M. (2013). Multinomial inverse regression for text analysis. *JASA*, **108**(503), 755–770.

The Apache Software Foundation (2013). Apache Lucene.

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

Wang, C. and Blei, D. M. (2013). Variational inference in nonconjugate models. *Journal of Machine Learning Research*.

Wolpert, D. and Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transaction on Evolutionary Computation*, **1**(1), 67–82.